UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Thiago Acórdi Ramos

PRESERVAÇÃO DO SIGILO E AUTENTICIDADE DE DOCUMENTOS ELETRÔNICOS POR LONGO PRAZO

Florianópolis
2011

Thiago Acórdi Ramos

PRESERVAÇÃO DO SIGILO E AUTENTICIDADE DE DOCUMENTOS ELETRÔNICOS POR LONGO PRAZO

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do grau de mestre em Ciência da Computação.

Ricardo Felipe Custódio, Dr.
Orientador

Florianópolis
2011

Catalogação na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

R175p Ramos, Thiago Acórdi
Preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo [dissertação] / Thiago Acórdi Ramos ; orientador, Ricardo Felipe Custódio. - Florianópolis, SC, 2011.
100 p.: il., grafs., tabs.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da computação. 2. Sigilo. 3. Autenticidade. 4. Documentos eletrônicos. I. Custódio, Ricardo Felipe. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 681

Thiago Acórdi Ramos

PRESERVAÇÃO DO SIGILO E AUTENTICIDADE DE DOCUMENTOS ELETRÔNICOS POR LONGO PRAZO

Esta dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

Florianópolis, 1 de Março de 2011

Mário Antonio Ribeiro Dantas, Dr.
Coordenador do PPGCC

Banca Examinadora:

Ricardo Felipe Custódio, Dr.
Orientador
Universidade Federal de Santa Catarina

Joni da Silva Fraga, Dr.
Universidade Federal de Santa Catarina

Lau Cheuk Lung, Dr.
Universidade Federal de Santa Catarina

Roberto Samarone dos Santos Araújo, Dr.
Universidade Federal do Pará

AGRADECIMENTOS

Meus mais sinceros agradecimentos a todos aqueles que colaboraram com este trabalho, de alguma forma. Inicialmente, agradeço aos meus pais Walter Arcelino Ramos e Taisa L. da S. Acórdi Ramos que não mediram esforços para propiciar o melhor para mim.

À minha namorada Caroline Maes, que além de namorada, foi certamente uma das maiores fontes de incentivo para a conclusão deste trabalho. Também pela disponibilidade de me acompanhar à Atenas na apresentação de um artigo e pelo planejamento da viagem. Ainda, por seu carinho e compreensão nos momentos de ausência.

Ao professor Dr. Ricardo Felipe Custódio, meu orientador, pelas oportunidades confiadas a mim, visando minha formação profissional. Os desafios apresentados construíram meu caminho pela área de segurança da informação, a qual sempre tive interesse e curiosidade.

Ao meu grande amigo, Nelson da Silva, com o qual tive oportunidade de compartilhar diversos desafios acadêmicos e pessoais. Muito da minha formação, profissional e pessoal, deve-se a nossas incontáveis horas de discussões e estudos.

Aos professores Dr. Lau Cheuk Lung e Dr. Roberto Samarone dos Santos Araújo e o amigo Jonathan Gehard Kholer. Suas colaborações proporcionaram a publicação do artigo submetido ao EuroPKI em minha primeira tentativa.

Aos amigos do LabSEC com os quais pude compartilhar e obter conhecimentos, além dos momentos de descontração e confraternização. Destaco a constante interação com os mestrandos Cristian Thiago Moecke, Jeandré Monteiro Sutil, Jonathan Gehard Kohler e Martín Augusto Gagliotti Vigil, além dos alunos de graduação Deise Luise Wrasse, Lucas Ferraro, Lucas Silveira, Lucila Alosilla e Maurício Simões de Oliveira. Em especial, gostaria de agradecer ao Dr. Jean Everson Martina, ao doutorando Marcelo Carlomagno Carlos e suas esposas por recepcionarem a mim e minha namorada em suas casas e pela convivência na Inglaterra.

Por fim, agradeço à Câmara Brasileira de Comércio Eletrônico (Camara-e.net) pelo financiamento das pesquisas contidas neste trabalho, além das oportunidades propiciadas, como publicação de um livro. Agradeço ainda aos parceiros do LabSEC com os quais tive oportunidade de trabalhar, dentre eles, o Instituto Nacional de Tecnologia da Informação (ITI), Colégio Notarial do Brasil (CNB), Softplan e BRy Tecnologia S.A.

RESUMO

Assinaturas digitais e carimbos do tempo são uma das formas de se preservar a autenticidade por longo prazo e já são empregados em diversas aplicações. O sigilo de documentos eletrônicos, por outro lado, é constantemente promovido apenas por controle de acesso. Igualmente, não se conhece sistemas com essa funcionalidade por longo prazo. Adicionalmente, necessita-se preservar em sigilo documentos eletrônicos assinados digitalmente, a exemplo dos atos processuais. Nesse sentido, verificou-se a existência de uma proposta unificando ambas as propriedades. Todavia, analisou-se tal abordagem e diversas deficiências foram constatadas. Assim, propôs-se dois protocolos para a preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo que aprimoram esse trabalho de modo a suprir as carências verificadas. Esses protocolos foram elaborados partindo de um protocolo base e adicionados outros mecanismos de modo a complementa-los. Avaliou-se os protocolos propostos relacionados a questão temporal e a resistência aos modelos de adversários considerados, de acordo com a literatura científica. Um protótipo foi desenvolvido no qual realizou-se testes e simulações. Os resultados obtidos da análise e implementação confirmam as informações teóricas e demonstram a possibilidade de implantação do protocolo em uma infraestrutura de longo prazo.

Palavras-chave: sigilo; autenticidade; longo prazo; documentos eletrônicos.

ABSTRACT

Digital signatures and timestamps are one way to preserve the long-term authenticity and are already employed in various applications. The secrecy of electronic documents, however, is generally promoted only by access control. Equally, there are no known systems with this functionality for long term. Additionally, there is need to preserve the secrecy of digitally signed electronic documents, like the procedural acts. In this sense, it was verified the existence of a proposal, unifying both properties. Nevertheless, this proposal was analysed and several deficiencies were found. Thus, it was proposed two protocols for the long-term preservation of secrecy and authenticity of electronic documents that improves this work in order to overcome the verified weaknesses. These protocols were developed starting from a base protocol and were added other mechanisms in order to complement them. It was evaluated the proposed protocols related to temporal issues and the resistance to models of adversaries considered in accordance with the scientific literature. A prototype was developed in which tests and simulations were conducted. The results obtained of the analysis and the implementation confirm the theoretical informations and demonstrates the possibility of deployment of the protocol in a long-term infrastructure.

Keywords: secrecy; authenticity; long-term; eletronic documents.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora
ACT	Autoridade de Carimbo do Tempo
AES	Advanced Encryption Standard
ASN.1	Abstract Syntax Notation One
CAdES	CMS Advanced Electronic Signatures
CDT	Certificado Digital Temporal
CNSEC	Central Notarial de Serviços Eletrônicos Compartilhados
DES	Data Encryption Standard
ERS	Evidence Record Syntax
ETSI	European Telecommunications Standards Institute
EuroPKI	European Workshop on Public Key Services, Applications and Infrastructures
ICP	Infraestrutura de Chaves Públicas
IDA	Information Dispersal Algorithm
ITI	Instituto Nacional de Tecnologia da Informação
LabSEC	Laboratório de Segurança em Computação
LCS	Laboratory for Computer Science
LTA	Long-Term Archive
LTANS	Long-Term Archive and Notary Services
MIT	Massachusetts Institute of Technology
MP	Medida Provisória
NIST	National Institute of Standards and Technology
OAIS	Open Archival Information System
OTP	One-time Pad
PAdES	PDF Advanced Electronic Signatures
PL	Projeto de Lei
RFC	Request For Comments
TJ-SP	Tribunal de Justiça do Estado de São Paulo
VSR	Verifiable Secret Redistribution
VSS	Verifiable Secret Sharing
XAdES	XML Advanced Electronic Signatures
XSS	Cross-site Scripting
xVSR	Extended Verifiable Secret Redistribution

SUMÁRIO

1 INTRODUÇÃO

O computador e a Internet já fazem parte do dia-a-dia e a importância dessas tecnologias na vida das pessoas só aumenta, trazendo vantagens como a desburocratização, celeridade e conectividade mundial. Com a desmaterialização dos documentos em papel para o meio eletrônico, a tendência é que a dependência do papel seja substituída pelo meio eletrônico. Exemplos dessa tendência são a crescente popularização de redes sociais, blogs e serviços de mensagem pessoal. Isso tem ocorrido na maioria dos países. Destaca-se que por documento eletrônico entende-se uma sequência de bytes, não importando a semântica desse documento.

No Brasil, por exemplo, a Medida Provisória (MP) 2.200-2 (BRASIL, 2001) que "*Institui a Infra-Estrutura de Chaves Públicas Brasileira – ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências*" tornou a assinatura digital equivalente à manuscrita. E a Lei 11.419 (BRASIL, 2006) que "*Dispõe sobre a informatização do processo judicial; altera a Lei no 5.869 (BRASIL, 1973), de 11 de janeiro de 1973 — Código de Processo Civil; e dá outras providências*" normatizou o meio eletrônico para o processo judicial. Assim, tem-se base na legislação para a desmaterialização dos processos por meios técnicos.

Uma rápida pesquisa ao sítio da câmara dos deputados[1] retorna alguns projetos de lei (PLs) que alteram leis para que determinada ação possa ser realizada em meio digital, por meio de assinaturas digitais.

Alguns serviços já são disponibilizados ao cidadão neste meio e a tendência é de crescimento. A Receita Federal[2] é o principal exemplo dessa desmaterialização. A declaração de imposto de renda, que antes era realizada em formulários em papel, passou para o meio eletrônico, inicialmente entregue por meio de disquetes, e hoje é entregue via Internet, podendo ser assinada digitalmente por certificado ICP-Brasil.

As serventias extrajudiciais – os cartórios – e o poder judiciário também seguem esforços na migração de suas bases de dados para o meio digital. Processo e peticionamento eletrônicos são hoje uma realidade.

Um subconjunto desses documentos é de caráter confidencial – sejam documentos privados ou processos que correm em segredo de justiça, por exemplo. Nos documentos em papel o conteúdo está atrelado ao substrato enquanto que os documentos eletrônicos são uma sequência de bytes que podem ser armazenadas em qualquer meio eletrônico.

O sigilo de documentos em papel pode ser obtido com um simples dobramento do papel ou inserindo-o em um envelope. Ao se utilizar um lacre, por exemplo de cera ou cola, caso este seja violado tem-se evidên-

[1] http://www.camara.gov.br

[2] http://www.receita.fazenda.gov.br

cias desse acontecimento. Ou seja, com os documentos em papel, por estarem em um meio físico, tem-se o controle da informação. No meio digital a informação pode ser facilmente copiada, visualizada e alterada, podendo nem deixar vestígios, ou serem de difícil rastreamento, o que torna o documento eletrônico, por si só, frágil em relação a confidencialidade da informação.

Um documento sigiloso, tanto em papel como eletrônico, pode ser mantido em um cofre ou sala a qual somente os interessados tenham acesso. Tal método é comum para documentos em papel, visto que já é necessário um espaço físico para armazená-los e desejável este controle. Com os documentos eletrônicos precisa-se de técnicas adicionais, pois normalmente estão armazenados em servidores conectados à Internet e qualquer outro computador conectado à rede pode alcançá-los. Essa exposição facilita o acesso de participantes ao documento de qualquer lugar, mas também o expõe a interessados não autorizados. A incerteza da segurança de sistemas, softwares e redes computacionais dificulta o sigilo de documentos eletrônicos dessa forma.

Faz-se necessário, portanto, o emprego de técnicas de segurança nos documentos a fim de torná-los sigilosos. Todavia, tal processo não é trivial visto que tais técnicas trazem novas complexidades no gerenciamento da informação. A criptografia é uma forma de tornar um documento eletrônico sigiloso de modo que o acesso ao documento não é negado, mas se copiado na forma cifrada, tem-se uma informação ininteligível. Mas o simples uso de criptografia não garante o sigilo, sendo necessário entender os mecanismos e processos envolvidos.

A preservação de documentos em papel é inerente ao meio, ou seja, a manutenção do papel, da tinta, entre outros. Há processos já consagrados de preservação desse meio, existindo documentos em papel de milhares de anos. A preservação de longo prazo de documentos eletrônicos por si só já é complexa dada a natureza eletrônica, obsolescência de tecnologias, inexistência de exatidão dos algoritmos de softwares, entre outros. A preservação do sigilo de documentos eletrônicos é ainda mais complexa.

Existem diferentes técnicas na literatura que promovem o sigilo de documentos eletrônicos que são eficazes. Entretanto, essa eficácia pode degradar ao longo do tempo e, assim, expor a informação sigilosa. Deve-se analisar os problemas que surgem quando o foco é o longo prazo e os ataques que podem comprometer o sigilo de documentos eletrônicos.

Adicionalmente, deve-se empregar técnicas para manutenção da autenticidade desses documentos eletrônicos que se adaptem as técnicas de sigilo por longo prazo adotadas. Ainda, promover a disponibilidade desses documentos sempre que requisitados. As assinaturas digitais já são empregadas como forma de autenticidade e os carimbos do tempo são empregados para manutenção dessas assinaturas por longo prazo. Tais características podem ser observadas no padrão brasileiro de assinatura digital (ITI, 2010b).

A junção das propriedades de autenticidade e confidencialidade

torna ainda mais complexa a preservação de documentos eletrônicos por longo prazo. Não há um padrão para a preservação dessas propriedades. Normalmente tais características são tratadas em separado e, muitas vezes, são até conflitantes de modo que ao se obter autenticidade não se consegue obter sigilo e vice-versa. Para tanto é necessário um estudo sobre as técnicas e trabalhos existentes a fim de encontrar uma resposta para essa questão.

O presente trabalho insere-se nesse contexto, buscando explorar as técnicas que promovem sigilo por longo prazo e que possam ser empregadas em conjunto com assinaturas digitais e carimbos do tempo que são as tecnologias existentes para preservação da autenticidade por longo prazo.

A proposta de um esquema com essas propriedades tem o intuito de elucidar e materializar as discussões realizadas, apontando os mecanismos e técnicas que sejam adequados para longo prazo. Tal proposta pode ser empregada como base para um sistema de preservação do sigilo e autenticidade de documentos eletrônicos, devendo-se, para tanto, agregar outras funcionalidades e necessidades não abordadas neste trabalho, mas que são de fundamental importância para a preservação de longo prazo.

1.1 OBJETIVOS

Descreve-se, nesta seção, os objetivos geral e específicos deste trabalho.

1.1.1 Objetivo Geral

Elaborar um protocolo com os seus respectivos mecanismos que possibilite a preservação do sigilo e da autenticidade de documentos eletrônicos por longo prazo.

1.1.2 Objetivos Específicos

- Realizar um levantamento das técnicas aplicadas à preservação do sigilo de documentos eletrônicos existentes na literatura científica e tecnológica, incluindo aquelas de longo prazo;
- Avaliar a aplicabilidade dessas técnicas em longo prazo de acordo com a literatura científica, comparando os benefícios e limitações de cada técnica;
- Discutir os ataques possíveis nas técnicas aplicáveis;
- Verificar por propostas existentes e avaliá-las em relação as técnicas, aplicabilidade e ataques discutidos;
- Propor um esquema para a preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo;

- Avaliar o esquema proposto e verificá-lo em relação as propostas existentes;

- Discutir sobre a proposta e a preservação de longo prazo.

1.2 METODOLOGIA

A partir da definição do problema e do levantamento das técnicas de sigilo conhecidas, verificou-se, de modo a comparar os benefícios e limitações de cada mecanismo, a aplicabilidade dessas técnicas para a preservação da confidencialidade de documentos eletrônicos. Do mesmo modo, analisou-se sobre os adversários e limitações das técnicas encontradas, principalmente, nas questões de longo prazo.

A comparação das técnicas bem como a discussão dos ataques deu-se baseada em informações, estudos, discussões e reflexões dos diversos trabalhos presentes na literatura específica, com enfoque na preservação de longo prazo, uma vez que os desafios aumentam com o passar do tempo. Por questões tecnológicas, a preservação por longo prazo é muito mais complexa que curto ou médio prazos.

Verificou-se a existência na literatura de um trabalho com o objetivo de unir autenticidade e confidencialidade de documentos eletrônicos. Percebeu-se, com base nos estudos realizados, que tal proposta possuía problemas e lacunas em relação ao que foi proposto.

Com uma melhor compreensão dos mecanismos criptográficos e ataques e sabendo-se que assinaturas digitais e carimbos do tempo são uma forma de preservação da autenticidade, elaborou-se dois protocolos nos quais são aplicadas técnicas para preservação da confidencialidade e autenticidade de documentos eletrônicos por longo prazo. O primeiro protocolo foi avaliado teórica e quantitativamente, e devido seus custos e limitações, foi aprimorado dando origem ao segundo protocolo. Analisou-se este protocolo com base na avaliação das técnicas por longo prazo, ataques, simulações e testes realizados com um protótipo elaborado.

Assumiu-se a hipótese de que a preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo era possível e praticável, sem que houvesse uma degradação da confidencialidade ou mesmo uma exposição da informação sigilosa, considerando-se as premissas adotadas.

1.3 JUSTIFICATIVA

O sigilo de documentos eletrônicos é de crescente interesse e importância devido à desmaterialização de documentos e processos que requerem confidencialidade. Além disso, tem-se a necessidade de preservação da autenticidade desses documentos, uma vez que a assinatura digital possui equiparação à manuscrita e, portanto, validade legal.

O Tribunal de Justiça de São Paulo (TJ-SP) utiliza o processo eletrônico, gerenciando todo o ciclo de vida dos atos processuais em

meio eletrônico, empregando assinaturas digitais e carimbos do tempo conforme padrões da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil). Entretanto, os processos que necessitam de sigilo estão protegidos apenas por controle de acesso, visto que não se conhece sistemas para a preservação do sigilo e autenticidade de documentos eletrônicos.

A escassez de literatura sobre confidencialidade de documentos eletrônicos, em relação à assinatura digital por exemplo, demonstra a dificuldade do tema, principalmente em longo prazo. A adição do requisito de preservação da autenticidade em conjunto com o sigilo torna o assunto ainda mais complexo, uma vez que normalmente tais propriedades são tratadas em separado.

O TJ-SP é apenas uma das entidades que seriam beneficiadas com este trabalho. Diversas outras entidades, como empresas e governo, precisam de uma solução para a preservação de longo prazo. Segredos industriais, militares e estratégias diversas são exemplos dessa necessidade.

Assim, este trabalho visa propor uma alternativa para a preservação da autenticidade e sigilo de documentos eletrônicos, contribuindo com a literatura científica dessa importante área que, a medida em que o mundo torna-se mais digital, torna-se cada vez mais necessária.

1.4 MOTIVAÇÃO

O Laboratório de Segurança em Computação (LabSEC) tem desenvolvido pesquisas, projetos e parcerias na área de segurança em computação, em especial em infraestrutura de chaves públicas (ICP) e certificação digital. Documentos eletrônicos é uma das linhas de pesquisa do LabSEC e diversos trabalhos já foram elaborados.

A tese de doutorado sobre a confiança no documento eletrônico (DIAS, 2004) levantou os requisitos de segurança de documentos em papel e os aplicou para os documentos eletrônicos. As dissertações sobre uma infraestrutura para datação de documentos eletrônicos (PASQUAL, 2001) e uma infraestrutura de armazenamento e recuperação segura de documentos eletrônicos (NOTOYA, 2002) são outros trabalhos que visam propiciar as mesmas condições dos documentos em papel para os eletrônicos.

A central notarial de serviços eletrônicos compartilhados (CNSEC) (SILVA et al., 2007) foi um estudo elaborado a pedido do Colégio Notarial do Brasil (CNB) para modernização e integração de cartórios, tratando todo o ciclo de vida dos documentos eletrônicos de forma segura.

Na linha de preservação de longo prazo desenvolveu-se um trabalho de conclusão de curso propondo um arcabouço para a preservação de longo prazo das propriedades de segurança de documentos eletrônicos (SILVA; RAMOS, 2007).

Ainda, tem-se diversos artigos científicos sobre documento eletrônico e certificação digital. Destacam-se a infraestrutura para liberação temporal de chaves (CUSTóDIO et al., 2007) e uma infraestrutura para

o arquivamento de longo prazo de documentos eletrônicos autênticos e sigilosos (RAMOS et al., 2011), que faz parte deste trabalho.

Participou-se também da elaboração do padrão brasileiro de assinatura digital, inicialmente em um estágio de um mês nas dependências do Instituto Nacional de Tecnologia da Informação (ITI) e, posteriormente, em palestras, pesquisas, consulta pública, grupos de trabalho, entre outros.

No decorrer do mestrado foram realizados dois projetos de sigilo para o Tribunal de Justiça de São Paulo (TJ-SP) – um para documentos e outro para mídias, os quais elevaram o interesse de pesquisa na área de sigilo de documentos eletrônicos, definido como tema da dissertação em acordo com o professor orientador.

Em tais projetos propôs-se uma infraestrutura para o sigilo de documentos eletrônicos e mídias digitais, visto que processos e audiências estão migrando para o meio digital naquele tribunal. Elaborou-se protocolos para o ciframento, deciframento e atualização da tecnologia criptográfica envolvida quando necessário. Vislumbrou-se também esquemas de *cache*, similar aqueles empregados em arquitetura de computadores, para aprimorar o desempenho da infraestrutura.

Os projetos elaborados levantaram a questão do sigilo para longo prazo, visto que a técnica de ciframento, empregada em ambos os projetos, pode ser quebrada ou mesmo degradar-se com o tempo. Como existem processos que devem correr em segredo de justiça é um requisito que a confidencialidade seja preservada pelo tempo que for necessário. Também existe o requisito de preservação da autenticidade por longo prazo, visto que as assinaturas digitais e carimbos do tempo são empregados para dar autenticidade aos atos processuais.

Encontrou-se, igualmente, um ambiente favorável a pesquisa no LabSEC, contando com infraestrutura, parcerias e recursos necessários para o desenvolvimento da pesquisa científica. A combinação desses fatores culminou na pesquisa e elaboração deste trabalho. Na linha de pesquisa de preservação por longo prazo tem-se duas dissertações em andamento, uma sobre a preservação de assinaturas digitais por longo prazo e a presente dissertação.

1.5 RESULTADOS ESPERADOS

Espera-se com este trabalho impulsionar discussões no Brasil para este importante tema – o sigilo de documentos eletrônicos, fomentando pesquisas, projetos e implementações, tal como tem acontecido com as assinaturas digitais (ITI, 2010b) e, mais recentemente, com os carimbos do tempo (ITI, 2010a).

O esquema esboçado pode ser usado como um ponto de partida para projetos e implementações de sistemas para preservação por longo prazo de documentos eletrônicos para as diversas entidades que possam vir a necessitar de uma infraestrutura com essas características. Um exem-

plo de entidade é a Receita Federal que hoje já emprega certificação digital em diversos processos, como a declaração de imposto de renda, mas que não deve dispor de soluções adequadas para preservação da autenticidade e sigilo desses documentos eletrônicos.

1.6 LIMITAÇÕES DO TRABALHO

A preservação de documentos eletrônicos por longo prazo envolve diversos aspectos dada as características desse tipo de documento.

O enfoque deste trabalho é com relação ao sigilo e autenticidade dos documentos eletrônicos em longo prazo (o que provê ainda preservação da integridade). Entretanto, outras características e problemas foram ou devem ser resolvidos a fim de solucionar todos os problemas da preservação digital por longo prazo.

Exemplos de problemas e dificuldades não abordadas incluem a obsolescência de formatos de arquivos, mídias, hardware e software; estratégias de preservação digitais como emulação e migração; a interpretabilidade de documentos e formatos de arquivos; segurança de sistemas operacionais, redes de computadores; heterogeneidade de softwares e mídias.

Por meio dos exemplos, percebe-se o quão complexa é a preservação digital. Existem trabalhos, discussões, projetos e implementações na literatura para esses problemas. Entretanto, ainda não se dispõe de uma solução definitiva.

O presente trabalho insere-se como uma pesquisa e discussão sobre o sigilo e autenticidade de documentos eletrônicos com foco nos mecanismos de segurança da informação e a preservação dessas propriedades por longo prazo.

1.7 ESTRUTURA DO TRABALHO

No capítulo 2 são definidos os conceitos básicos necessários para compreensão do trabalho. Lista-se os princípios de Kerckhoffs os quais norteiam o desenvolvimento das técnicas criptográficas modernas. Faz-se um levantamento das técnicas de sigilo disponíveis na literatura. Descreve-se as técnicas de assinatura digital e carimbos do tempo como forma de preservação da autenticidade. Ainda, apresenta-se os modelos de adversários que serão considerados para avaliação das técnicas.

Com base no capítulo anterior, os trabalhos relacionados encontrados na literatura científica são descritos no capítulo 3, detalhando-se com maior riqueza as propostas de fundamental importância para este trabalho. Estuda-se as técnicas de compartilhamento de segredo e variantes, além de protocolos de redistribuição de segredo. Ainda, descreve-se a sintaxe de registro de evidência e cita-se alguns sistemas de arquivamento. Faz-se uma visão geral de uma proposta – a arquitetura de referência.

A seguir, no capítulo 4, são avaliados as técnicas e trabalhos descritos nos capítulos 2 e 3 em relação à adequabilidade desses por longo prazo. Analisa-se tais técnicas, também, relativamente aos modelos de adversários que podem comprometer os mecanismos descritos. Ainda, avalia-se a arquitetura de referência proposta na literatura científica como infraestrutura de sigilo e autenticidade de documentos eletrônicos com base nas análises realizadas.

Elabora-se dois protocolos para a preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo no capítulo 5, levando-se em consideração os estudos realizados nos capítulos anteriores. Avalia-se, também, o protocolo preliminar em relação aos seus custos e limitações. Com base nessa análise, propõe-se um novo protocolo de modo a aprimorar os resultados obtidos preliminarmente. Descreve-se as etapas do novo protocolo e as etapas da sintaxe do registro de evidência para esse protocolo.

O capítulo 6 tem como objetivo avaliar o novo protocolo proposto no capítulo 5 em relação as deficiências averiguadas na arquitetura de referência, estas verificadas no capítulo 4. Descreve-se, também, os testes e simulações realizados por meio da implementação de partes do novo protocolo. Conclusões obtidas com os testes e simulações também são apresentadas. Discute-se, ainda, os protocolos propostos, a preservação de longo prazo e algumas ideias de modo a melhorar as propostas realizadas.

Por fim, faz-se as considerações finais no capítulo 7.

2 CONCEITOS BÁSICOS

2.1 INTRODUÇÃO

Antes de aprofundar nos trabalhos relacionados faz-se necessário descrever algumas definições e conceitos de segurança, técnicas criptográficas para sigilo e autenticidade, além de formas de ameaça à segurança da informação.

Considera-se que o leitor já tenha conhecimentos básicos na área de criptografia e, caso não os possua, recomenda-se a leitura prévia de capítulos introdutórios em literaturas clássicas, como Schneier (1996) e Stallings (2002).

Na seção 2.2 são descritas definições e conceitos de segurança que serão utilizados ao longo deste trabalho. Enumera-se, na seção 2.3, os princípios de Kerckhoffs que norteiam o desenvolvimento de algoritmos criptográficos. Listam-se as técnicas encontradas na literatura que promovem o sigilo da informação, um dos focos deste trabalho, na seção 2.4. Na seção seguinte, 2.5, são mostrados os modelos de adversários que serão considerados para avaliação das técnicas descritas. A seção 2.6 descreve as técnicas de assinatura digital e carimbo do tempo como forma de preservação da autenticidade. Por fim, faz-se as conclusões deste capítulo na seção 2.7.

2.2 DEFINIÇÕES DE SEGURANÇA

Inicialmente, define-se as propriedades de segurança, de acordo com a RFC 4949 (SHIREY, 2007, tradução nossa):

autenticidade "a propriedade de ser genuíno e capaz de ser verificável e confiável"; Neste trabalho, considera-se que autenticidade provê integridade[1].

confidencialidade "a propriedade que o dado não está divulgado para entidades do sistema a menos que tenham sido autorizadas a conhecer o dado". Neste trabalho, utiliza-se sigilo como sinônimo de confidencialidade;

integridade "a propriedade que o dado não foi alterado, destruído ou perdido de uma maneira não autorizada ou acidental".

Em criptografia é um desafio provar rigorosamente a segurança de sistemas e protocolos. De acordo com Maurer (1993), para provar a se-

[1] Segundo Menezes, Oorschot e Vanstone (1997, tradução nossa), "[...] se uma mensagem está modificada, a fonte foi alterada".

gurança de um sistema criptográfico é necessário: uma definição de segurança (ou de quebra do sistema) e asserções sobre quanto de informação e quanto poder computacional dispõe um adversário. Normalmente, pelos princípios de Kerckhoffs (ver seção 2.3), assume-se que o adversário possui completo conhecimento do sistema e pode receber todas as informações transmitidas.

Em relação à consideração do poder computacional de um adversário, presumem-se dois tipos de conceitos, de acordo com Maurer (1993, tradução nossa):

Computacionalmente seguro "o sistema [...] é seguro se o adversário possui recursos computacionais razoavelmente limitados";

Informação teoricamente segura "o sistema [...] é seguro mesmo se o adversário possuir recursos computacionais ilimitados". Também chamado de incondicionalmente seguro (MAURER, 1999).

Schneier (1996, tradução nossa), similarmente, descreve tais conceitos como:

Segurança Condicional "Um algoritmo é considerado computacionalmente seguro (as vezes chamado forte) se não puder ser quebrado com os recursos disponíveis, tanto corrente quanto futuros"; e

Segurança Incondicional "Um algoritmo é incondicionalmente seguro se, não importa quanto do texto cifrado o criptoanalista dispor, não há informação suficiente para recuperar o texto plano".

Segundo Maurer (1993, tradução nossa), para o primeiro conceito – computacionalmente seguro, existem dois problemas. O primeiro é em relação ao modelo computacional, "[...] não sendo claro se algum modelo é suficientemente geral". O segundo diz respeito a teoria da complexidade, a qual "[...] não consegue fornecer prova de qualquer limite mínimo razoável para qualquer problema razoável ou modelo computacional".

A teoria da informação, descrita por Shannon (1948), definiu meios para provar a segurança de um sistema, mesmo em presença de adversários com recursos computacionais ilimitados, com base nessa teoria. Foi proposto que um sistema é perfeito se o texto cifrado não provê qualquer informação sobre o texto plano (o "[...] texto cifrado e o texto plano são estatisticamente independentes") (MAURER, 1993, tradução nossa).

Faz-se necessário, também, conceituar a questão temporal. Tais conceitos não são simples de se estabelecer uma vez que depende da situação. Schneier (1996, tradução nossa), por exemplo, diz que algumas perguntas devem ser feitas para determinar o nível de segurança necessário: "quanto vale a informação? Por quanto tempo a informação precisa ser mantida segura? Quais os recursos dos adversários?"

De modo a estabelecer um parâmetro, para este trabalho define-se os seguintes conceitos:

curto prazo informações que necessitam de sigilo por um curto período de tempo, por exemplo, ofertas em licitações;

médio prazo informações cujo sigilo requer um período médio de tempo, por exemplo, a guarda de comprovantes[2];

longo prazo informações que requerem sigilo por um longo período de tempo, por exemplo, um testamento[3];

prazo indeterminado são aquelas informações sigilosas que não possuem limite de tempo para guarda, por exemplo, documentos militares e segredos industriais.

Este trabalho tem como foco os dois últimos períodos de tempo (longo e indeterminado) visto que para curto e médio prazos já existem soluções adequadas na literatura científica. Adicionalmente, em relação ao tempo, ambos serão considerados sinônimos.

2.3 PRINCÍPIOS DE KERCKHOFFS

Kerckhoffs (1883) enumerou alguns princípios, hoje conhecidos como Princípios de Kerckhoffs, os quais norteiam a criptografia moderna. São eles:

1. *Le système doit être matériellement, sinon mathématiquement, indéchiffrable;*
2. *Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi;*
3. *La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants;*
4. *Il faut qu'il soit applicable à la correspondance télégraphique;*
5. *Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes;*
6. *Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.*

[2] Conforme a lei 5.172 (BRASIL, 1966), esse prazo é cerca de cinco anos.

[3] Apesar da data de abertura ser indeterminada, uma pessoa não vive por duzentos anos, por exemplo.

Em geral, os princípios dizem que: o sistema criptográfico deve ser indecifrável; pode ser conhecido pelo adversário; deve ser fácil de utilizar, atualizar ou modificar as chaves pelos diferentes participantes; e deve ser portátil e seu uso não deve exigir ajuda de várias pessoas.

Tais princípios foram fundamentais à criptografia moderna visto que a segurança por obscuridade foi deixada de lado em favor da publicação dos algoritmos, o que aumentou consideravelmente a segurança dos sistemas criptográficos. A competição[4] para encontrar o substituto do algoritmo *Data Encryption Standard* (DES), elaborada pelo *National Institute of Standards and Technology* (NIST), é um exemplo desses princípios. O algoritmo simétrico eleito (*Rijndael*) como *Advanced Encryption Standard* (AES) foi adotado somente após diversas rodadas, discussões e revisões, realizadas a partir das especificações dos algoritmos propostos divulgadas publicamente. O AES ainda é o algoritmo simétrico recomendado.

Em Novembro de 2007 o NIST lançou uma competição pública[5] para o desenvolvimento de um novo algoritmo de resumo criptográfico, que se espera ser concluída em 2012, constituindo em mais um exemplo dos princípios descritos acima.

2.4 TÉCNICAS DE CONFIDENCIALIDADE

As técnicas de confidencialidade das informações em meio eletrônico podem ser classificadas em sete diferentes categorias: controle de acesso, obscuridade, esteganografia, ciframento, dispersão da informação, compartilhamento de segredo e técnicas temporais.

2.4.1 Controle de Acesso

Talvez o mecanismo mais simples e a forma mais difundida de manutenção do sigilo de documentos eletrônicos seja o controle de acesso, visto ser análogo à forma convencional de sigilo de documentos em papel. Consiste no controle dos recursos, nesse caso o documento eletrônico, contra acesso não autorizado.

Segundo Shirey (2007, tradução nossa), é “um processo pelo qual o uso de recursos do sistema é controlado de acordo com uma política de segurança e é permitido somente por entidades autorizadas [...] conforme aquela política".

Tipos de controle de acesso incluem discricionário, obrigatório e baseado em papéis. De acordo com a RFC 4949 (SHIREY, 2007, tradução nossa), o termo discricionário é “[...] porque uma entidade pode ter direito de acesso garantido a um recurso tal que uma entidade pode por sua própria vontade habilitar outras entidades a acessar o recurso". O termo

[4]Fonte: `http://csrc.nist.gov/archive/aes/index2.html#overview`
[5]Fonte: `http://www.nist.gov/itl/csd/ct/hash_competition.cfm`

obrigatório é no sentido contrário do discricionário, “[...] porque uma entidade que possui liberação para acessar um recurso não é permitida, apenas por sua vontade própria, de habilitar outra entidade para acesso a esse recurso". Baseado em papéis refere-se ao controle de acesso no qual “entidades do sistema que são identificadas e controladas possuem posições funcionais em uma organização ou processo".

2.4.2 Segurança por Obscuridade

A segurança por obscuridade é outra ideia que surge ao se pensar em confidencialidade. Se o atacante não souber como a confidencialidade é feita ele não saberá como desfazê-la.

Um exemplo comum desta técnica são as licenças (seriais) de software que, normalmente, são geradas com alguma fórmula de conhecimento exclusivo dos desenvolvedores.

2.4.3 Esteganografia

De acordo com Jamil (1999, grifo do autor, tradução nossa), “Derivado da língua grega, a palavra esteganografia significa literalmente *escrita oculta*". Ainda em conformidade com o referido autor, modernamente esteganografia significa “[...] a arte (assim como a ciência) da comunicação em uma forma oculta".

Anderson e Petitcolas (1998, tradução nossa) ainda acrescentam que “A incorporação é tipicamente parametrizada por uma chave; sem o conhecimento dessa chave [...] é difícil para uma terceira parte detectar ou remover o material incorporado".

Assim, a técnica de esteganografia tem como objetivo inserir uma mensagem secreta, de alguma forma, em meio à outra informação. Exemplos dessa técnica incluem a seleção de letras em palavras, tinta invisível ou bits menos significativos de imagens (STALLINGS, 2002).

2.4.4 Ciframento

As técnicas modernas de ciframento podem ser dividas em algoritmos simétricos e assimétricos. Entretanto, em se tratando de confidencialidade de documentos eletrônicos, algoritmos simétricos são a escolha comum uma vez que são mais eficientes, e possuem maior longevidade que os assimétricos. De acordo com o estudo ECRYPT II (2010), uma chave simétrica de 128 bits é equivalente a uma chave assimétrica de 3248 bits e a uma curva elíptica com chave de 256 bits.

Mecanismos clássicos – como cifrador de César e máquina de rotores, ou técnicas de substituição e transposição – bases dos cifradores modernos, não serão tratados neste trabalho. Esse assunto possui vasta bibliografia, incluindo livros texto, a exemplo de Stallings (2002) e Denning (1982).

> The increased use of computer and communications systems by industry has increased the risk of theft of proprietary information. Although these threats may require a variety of countermeasures, encryption is a primary method of protecting valuable electronic information. (STALLINGS, 2002, p.26).

O uso do ciframento simétrico para sigilo de documentos eletrônicos requer a gerência das chaves utilizadas visto que, se a chave for encontrada, o documento sigiloso pode ser decifrado. Assim, são necessários mecanismos para o gerenciamento do ciclo de vida dessas chaves criptográficas uma vez que a confidencialidade de tais documentos depende da proteção e sigilo dessas chaves.

O gerenciamento de chaves é comumente abordado, entre outros meios, por controle de acesso, armazenamento em hardware criptográfico, ciframento com uma frase secreta (*password phrase*) e compartilhamento de segredo.

O *one-time pad* (OTP) consiste no ciframento, por meio de uma operação de ou-exclusivo, entre uma informação e uma chave aleatória com, pelo menos, o mesmo tamanho da informação plana.

2.4.5 Dispersão da Informação

Algoritmo de dispersão da informação (*Information Dispersal Algorithm* – IDA) é um método que divide um arquivo em n partes de forma que possa ser reconstruído a partir de um subconjunto dessas partes. Rabin (1989) elaborou uma abordagem na qual propôs a divisão de um arquivo F de tamanho $L = |F|$ em n partes F_i, $1 \leq i \leq n$, cada parte de tamanho $|F_i| = \frac{L}{m}$, tal que quaisquer m partes são suficientes para reconstruir F.

A classe de algoritmos de dispersão da informação:

> [...] pode ser vista como pertencente ao campo de códigos de correção de erro (*error correction codes*), na qual bits extras são adicionados à mensagem criando um bloco de modo que, mesmo com a ocorrência de k erros dentro do bloco, a mensagem ainda pode ser reconstruída. (RABIN, 1989, tradução nossa).

2.4.6 Compartilhamento de Segredo

De acordo com Shamir (1979), o compartilhamento de segredo (*Secret Sharing*) constitui um mecanismo no qual uma informação D é dividida em n partes $(D_1, D_2, ..., D_n)$ de modo que :

1. com quaisquer m ou mais D_i partes calcula-se facilmente D;

2. o conhecimento de $m-1$ ou menos D_i partes não revela qualquer informação sobre D.

Tal esquema é chamado um esquema de limiar (m, n). As n partes são distribuídas por até n participantes de modo que é necessária a colaboração de pelo menos $m \leq n$ partes para a reconstrução da informação original (o segredo). Os trabalhos de Shamir (1979) e Blakley (1979), independentemente, propuseram tal técnica.

Um caso particular do compartilhamento de segredo é quando $m = n$. Nesse caso, tem-se um *one-time pad*, com a chave compartilhada por n participantes.

2.4.7 Técnicas Temporais

As técnicas temporais são mecanismos para liberação programada de um segredo em um instante de tempo futuro. May (1993) descreveu primeiramente essa técnica com o uso de uma terceira parte confiável. Duas formas foram apresentadas: o armazenamento e a liberação do documento; e o ciframento do documento, manutenção da chave em sigilo e posterior liberação dessa chave.

Rivest, Shamir e Wagner (1996) criaram um mecanismo que produz a chave de deciframento a partir da resolução de um problema em tempo conhecido. Tal mecanismo foi criado de modo que não pudesse ser paralelizado ou acelerado com o uso de outros computadores, ou com o aumento do poder computacional. Um exemplo dessa técnica, embarcado em um sistema selado, é a capsula do tempo LCS35 criada por Rivest[6]. Espera-se que o segredo seja revelado por volta do 70º aniversário do *Laboratory for Computer Science* (LCS) do *Massachusetts Institute of Technology* (MIT), no ano de 2033.

Em outro exemplo dessas técnicas, Custódio et al. (2007) propuseram uma infraestrutura para liberação temporal de chaves. Tal infraestrutura é composta por duas autoridades de certificação temporal (AC Temporal), uma on-line e outra off-line, uma autoridade de carimbo do tempo (ACT) e módulos de ciframento, além de gestão de pessoal, como administradores e operadores.

> A autoridade de certificação temporal [...] gera um par de chaves criptográficas assimétricas. A chave privada é armazenada e mantida segura até uma data futura especificada quando é publicada. A chave pública é inserida em um documento eletrônico chamado certificado digital temporal (CDT) como especificado pelo padrão X.509v3. Usuários podem usar esse certificado digital para cifrar documentos. Uma vez cifrados, o documento e seu conteúdo somente se-

[6] http://people.csail.mit.edu/rivest/lcs35-puzzle-description.txt

> rão conhecidos quando a chave privada for publicada. (CUSTóDIO et al., 2007, tradução nossa).

A diferença entre as AC Temporal on-line e off-line é em relação a geração do par de chaves criptográficas, respectivamente, para curto e longo prazos. A autoridade on-line é também a interface com os usuários do serviço, recebendo requisições e disponibilizando as chaves criptográficas. A ACT é utilizada para provimento de data e hora para os documentos eletrônicos (ver seção 2.6.2). Os módulos de ciframento, por sua vez, realizam as operações criptográficas em hardware e ambiente seguros.

No certificado temporal estão informações como estratégias e data de divulgação. Após a geração do CDT, este é disponibilizado pela AC Temporal e qualquer entidade que deseje ter a abertura do seu documento naquela data pode utilizar esse certificado. Ou seja, a mesma chave pública é compartilhada por qualquer entidade que queira manter sigilosos documentos até a data contida no certificado. Após a expiração, que é a data de divulgação, o CDT é publicado pela autoridade, juntamente com a chave privada correspondente. Assim, as entidades podem acessar a AC Temporal, obter a chave privada para, então, decifrar o documento.

2.5 MODELOS DE ADVERSÁRIOS

A conceituação dos modelos de adversários é necessária para o entendimento das informações e discussões ao longo deste trabalho. Os modelos aqui relacionados são voltados às técnicas de confidencialidade e longo prazo.

Os modelos de adversários, no que tange o longo prazo e as técnicas de criptografia para confidencialidade (ciframento e compartilhamento de segredo), podem ser divididos nas seguintes categorias:

2.5.1 Adversários Ativos

Os adversários ativos (ou bizantinos) são aqueles que possuem completo acesso à infraestrutura, podendo corromper dados ou estados, alterar ou repetir mensagens, realizar escutas nos canais de comunicação, monitorar eventos, obter informações da rede, entre outras (WONG; WANG; WING, 2002).

2.5.2 Adversários Móveis

Wong, Wang e Wing (2002) definem adversários móveis (ou dinâmicos) como adversários que comprometem servidores progressivamente e se não impedidos, eventualmente comprometerão participantes (servidores) suficientes para revelar o segredo.

Como o objetivo é manter a informação sigilosa por longo prazo, o atacante tem todo esse período para acometer o sistema. Os ataques podem ser espaçados na linha do tempo de modo que seja o mais discreto

possível, possivelmente não disparando alarmes de sistemas de detecção de intrusão. Outra dificuldade diz respeito ao histórico de ataques, visto ser necessário manter um número suficiente de partes seguras em esquemas de segredo compartilhado (STORER; GREENAN; MILLER, 2006). No trabalho de Storer, Greenan e Miller (2006) é chamado de ataques lentos (*slow attacks*), mas se trata do mesmo conceito.

Esse tipo de ataque é especialmente comprometedor quando utilizada a técnica de compartilhamento de segredo. No exemplo de Storer, Greenan e Miller (2006), suponha um arquivo protegido usando um esquema $(3, 5)$ no qual um arquivo é divido em cinco partes, três dessas necessárias pra reconstruir o arquivo. Suponha que um atacante comprometeu o sistema e conseguiu uma das partes. Uma década depois o mesmo atacante obtém uma segunda parte. Pela natureza do segredo compartilhado, o atacante não obteve qualquer informação sobre o dado. Entretanto, o sistema deve lidar com o fato de que o atacante está fazendo progresso. Com a obtenção de somente mais uma parte, o segredo (o arquivo) será revelado. Do mesmo modo, o adversário pode excluir partes da informação ao invés de obtê-la e, assim, fazer com que o segredo seja perdido.

O exemplo anterior revela um problema que surge ao introduzir a técnica de compartilhamento de segredo. Um sistema que deseje utilizá-la deve manter um histórico de comprometimentos e lidar com essas situações, realizando ações corretivas para minimizar tal brecha do sistema.

2.5.3 Trapaceiro

O trapaceiro (*cheater*) é descrito no exemplo de He e Dawson (1998): Supondo um esquema (m, n) de compartilhamento de segredo e que m participantes mostram suas partes, um a um, sem um protocolo. Após ter visto as outras $m-1$ partes, o último participante pode se recusar a mostrar sua parte ou entregar uma parte falsa. No fim, este participante (o trapaceiro) pode reconstruir o segredo, mas os demais participantes não.

Os esquemas tradicionais de compartilhamento de segredo não possuem mecanismos contra trapaceiros. É necessário, portanto, resolver o problema da reconstrução justa do segredo, sendo que todos os participantes tenham a mesma quantidade informação revelada.

2.6 TÉCNICAS DE AUTENTICIDADE

Em relação as técnicas de autenticidade, descreve-se a assinatura digital como meio de promover essa propriedade. Além disso, apresenta-se o carimbo do tempo, inicialmente proposto como uma forma de datação, mas que pode ser empregado na preservação de assinaturas digitais por longo prazo.

2.6.1 Assinatura Digital

Diffie e Hellman (1976) conceituaram a criptografia de chaves públicas, ou assimétrica, com o objetivo de solucionar o problema do compartilhamento de uma mesma chave na criptografia simétrica. Tal proposta baseia-se no uso de duas chaves complementares tal que a operação realizada por uma somente pode ser revertida pela outra. Todavia, mesmo complementares, a derivação de uma das chaves a partir do conhecimento da outra é impraticável. Assim, uma das chaves é tornada pública enquanto a outra é mantida em sigilo (privada).

Rivest, Shamir e Adleman (1978) propuseram uma implementação prática para obter assinaturas digitais a partir dos conceitos propostos por Diffie e Hellman (1976). Esse método tornou-se conhecido como RSA[7].

As assinaturas digitais são um meio de promover a autenticidade de objetos eletrônicos. Aplicação direta do ciframento assimétrico, obtém-se o resumo criptográfico do objeto a ser autenticado e, utilizando a chave privada S_k, cifra-se esse resumo. Para verificação dessa assinatura, utiliza-se a chave pública correspondente S_u para decifrar o resumo cifrado e, a partir da obtenção do resumo criptográfico do objeto, compara-se ambos os resumos; caso coincidam, comprova-se a autenticidade do objeto.

Formatos distintos para assinaturas digitais foram propostos e aprimorados no decorrer do tempo. Os padrões CAdES (ETSI, 2009a), XAdES (ETSI, 2009c) e PAdES (ETSI, 2009b) do *European Telecommunications Standards Institute* (ETSI) constituem formatos de assinatura digital projetados para preservação da autenticidade por longo prazo por meio de carimbos do tempo.

Diferentemente das assinaturas manuscritas, as digitais perdem sua segurança ao longo do tempo quando os algoritmos ou parâmetros criptográficos empregados tornarem-se fracos; ou o certificado de alguma autoridade certificadora do caminho de certificação do signatário expirar ou for revogado; ou ainda quando o algoritmo de resumo criptográfico, utilizado na assinatura e ou certificados, tornar-se inseguro.

2.6.2 Carimbo do Tempo

Haber e Stornetta (1991) propôs duas formas de datação de um documento digital: por vinculação (*linking*) e confiança distribuída (*distributed trust*). A primeira consiste no encadeamento de resumos criptográficos a medida em que os clientes requisitarem carimbos do tempo. A precedência de um documento no encadeamento indica que esse documento é anterior ao atual. Já na segunda forma o cliente requisita para diversos receptores o carimbo do tempo, recebendo uma mensagem assinada por cada receptor contendo a data da requisição. Considera-se que exista uma maioria honesta de receptores e, assim sendo, obtém-se uma

[7] A primeira letra do sobrenome de seus inventores.

data confiável.

> Suponha que tenhamos duas implementações de carimbo do tempo, e que há uma razão para acreditar que a primeira implementação logo será quebrada. Então certificados emitidos usando a implementação antiga podem ser renovados usando a nova implementação. Considere um certificado de carimbo do tempo criado usando a implementação antiga que é carimbado com a nova implementação antes que a antiga seja quebrada. Antes da quebra da implementação antiga, o único modo de criar um certificado era por meios legítimos. Assim, carimbando o certificado propriamente dito com a nova implementação, tem-se evidência que não apenas aquele documento existia antes da data do novo carimbo, mas que o documento existia na data especificada no certificado original. (HABER; STORNETTA, 1991, tradução nossa).

Uma forma difundida de implementação do carimbo do tempo foi proposta na RFC 3161 (ADAMS et al., 2001) a qual descreveu um formato para o carimbo do tempo e um serviço de datação – a autoridade de carimbo do tempo (ACT).

> O papel da ACT é carimbar um dado para estabelecer evidência, indicando que esse dado existia antes de um determinado tempo. Isso pode então ser usado, por exemplo, para verificar que uma assinatura digital foi aplicada a uma mensagem antes de o certificado correspondente ser revogado, permitindo assim o uso do certificado de chave pública para verificação de assinaturas criadas antes do instante de revogação. (ADAMS et al., 2001, tradução nossa).

Aposto em uma assinatura digital, o carimbo do tempo atesta que essa assinatura já existia na data contida nesse carimbo. Além da função de datação, os carimbos do tempo também são empregados para a preservação de assinaturas digitais por longo prazo, renovando-os quando necessário.

2.7 CONCLUSÃO

Neste capítulo, descreveu-se definições e conceitos de segurança que serão utilizados ao longo deste trabalho. As definições temporais foram elaboradas e exemplificadas visto que não há um senso comum.

Enumerou-se os princípios de Kerckhoffs que norteiam o desenvolvimento de algoritmos criptográficos. Listou-se as técnicas encontradas na literatura que promovem o sigilo da informação. Tais técnicas foram classificadas em sete categorias, de acordo com suas características.

Buscou-se as técnicas que, de alguma forma, promovessem a confidencialidade da informação.

Foram mostrados os modelos de adversários que serão considerados para avaliação das técnicas descritas. Percebeu-se que adversários móveis e ataques referem-se ao mesmo tipo de adversário.

Por fim, descreveu-se as técnicas de assinatura digital e carimbo do tempo como forma de preservação da autenticidade.

No capítulo 3 serão apresentados os trabalhos relacionados que utilizam as informações descritas neste capítulo.

3 TRABALHOS RELACIONADOS

3.1 INTRODUÇÃO

Nesta capítulo serão revisados os trabalhos disponíveis na literatura que de alguma forma estão relacionados à presente dissertação. Tais trabalhos estão organizados em três grandes grupos: técnicas de compartilhamento, registro de evidência e sistemas de arquivamento. Descreve-se as principais características, contribuições e limitações de cada um desses trabalhos.

O primeiro grupo é composto pelas seguintes técnicas de confidencialidade: compartilhamento de segredo (seção 3.2), compartilhamento de segredo verificável (seção 3.3), compartilhamento de segredo proativo (seção 3.4) e protocolos de redistribuição de segredo (seção 3.5).

A sintaxe de registro de evidência, descrita na seção 3.6, compõe o segundo grupo, relacionado à autenticidade da informação.

O último grupo, composto pelos sistemas de arquivamento, é contemplado na seção 3.7. Adicionalmente, faz-se uma visão geral de um desses sistemas – a arquitetura de referência, na seção 3.8.

Por fim, faz-se as conclusões deste capítulo na seção 3.9.

3.2 COMPARTILHAMENTO DE SEGREDO DE SHAMIR

O compartilhamento de segredo foi previamente conceituado na seção 2.4.6. Nesta seção, descreve-se o método de Shamir (1979), pois a abordagem de Blakley (1979) é menos eficiente na questão de espaço: enquanto as partes em Shamir são tão grandes quanto o segredo, em Blakley elas são m vezes esse tamanho – onde m é o limiar (*threshold*).

A técnica de compartilhamento de segredo pode ser divida em três etapas: a construção do polinômio, o cálculo das partes e a reconstrução do polinômio. A seguir serão descritas cada uma dessas etapas e, posteriormente, traz-se um exemplo de aplicação da técnica.

Um modo de compartilhar um segredo entre n participantes, sendo necessária a colaboração de pelo menos m deles para reconstrução do segredo, é escolhendo-se um polinômio secreto f, de ordem $m-1$, onde a_0 (termo independente) é o segredo s que se quer compartilhar e os demais coeficientes são escolhidos aleatoriamente. Assim, o polinômio f é definido como:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{m-1} x^{m-1} \tag{3.1}$$

Nota-se que, quando $x = 0$, $y = f(0) = s$, ou seja, o segredo

s compartilhado. Os cálculos no computador são normalmente realizados por aritmética modular, utilizando um número primo p, formando um corpo finito $\mathbb{Z}_p$. Deve-se escolher um primo p maior que s e n. Os coeficientes são escolhidos aleatoriamente entre $[0, p)$ e os valores de $s_1, ..., s_n$ são computados módulo p.

O cálculo das partes do segredo consiste na avaliação do polinômio f nos diferentes pontos, ou seja, o cálculo dos valores $y_i = f(x_i)$ para $i = 1, \cdots, n$. Os n pontos (x_i, y_i) obtidos desse cálculo constituem as partes do segredo s.

Para reconstruir o segredo s, aplica-se uma técnica de interpolação de polinômios, coletando pelo menos m das n partes. É comum atribuir os valores de x_i como o identificador do participante, por exemplo, $x_1 = 1$. Assim, necessita-se apenas que cada participante i armazene o valor de y_i. Ao longo deste trabalho será adotada essa convenção.

Uma das técnicas mais conhecidas para interpolação de polinômios é a interpolação de Lagrange, que será utilizada nos exemplos deste trabalho. Por tal técnica, tem-se que:

$$f(x) = \sum_{i \in \mathcal{B}} y_i l_i, \text{ tal que } l_i(x) = \prod_{\substack{j \in \mathcal{B}, \\ j \neq i}} \left(\frac{x - x_j}{x_i - x_j}\right)$$

> O conjunto $\mathcal{B}$ denota um conjunto admissível. Um conjunto admissível é um conjunto que contém o número limiar de servidores (e consequentemente partes) de modo que o segredo pode ser reconstruído. Na formulação o limiar é m, então $\mathcal{B}$ contém m servidores (e consequentemente partes). (GUPTA; GOPINATH, 2006, tradução nossa).

Para exemplificar, suponha que se queira compartilhar o segredo $s = 42$ entre três participantes, sendo necessária a participação de pelo menos dois deles para remontar o segredo ($m = 2, n = 3$ e o polinômio $f(x)$ é de grau $m - 1 = 1$). Assim, $a_0 = 42$ é o segredo.

Seja $a_1 = 51$ um número gerado de forma aleatória. Escolheu-se $p = 53$ o tamanho do corpo. Então, $f(x) = a_0 + a_1 x \pmod p = 42 + 51x \pmod{53}$, tal que $f(0) = 42$. Calcula-se então os $n = 3$ pontos da função $f(x)$:

$$s_1 = f(1) = 42 + 51 = 40 \pmod{53}$$

$$s_2 = f(2) = 42 + 102 = 38 \pmod{53}$$

$$s_3 = f(3) = 42 + 153 = 36 \pmod{53}$$

Os três pontos obtidos são: $(x_0 = 1, y_0 = 40); (x_1 = 2, y_1 = 38); (x_2 = 3, y_2 = 36)$. Cada ponto deve ser armazenado por cada um

dos três participantes. Com a interpolação destes pontos, recupera-se o segredo. Aplicando Lagrange, tem-se:

$$l_0 = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} = \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} = \frac{1}{2}x^2 - \frac{5}{2}x + 3$$

$$l_1 = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} = \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} = -x^2 + 4x - 3$$

$$l_2 = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} = \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} = \frac{1}{2}x^2 - \frac{3}{2}x + 1$$

$$f(x) = y_0 \cdot l_0 + y_1 \cdot l_1 + y_2 \cdot l_2$$

$$f(x) = 40(\frac{1}{2}x^2 - \frac{5}{2}x + 3) + 38(-x^2 + 4x - 3) + 36(\frac{1}{2}x^2 - \frac{3}{2}x + 1) \pmod{53}$$

$$f(x) = 51x + 42 \pmod{53}$$

$$f(0) = 42 = s$$

Percebe-se no exemplo que pela reconstrução obtém-se o mesmo polinômio que gerou tais pontos. Destaca-se que no exemplo foram utilizados três pontos para interpolação. Entretanto, somente dois deles $(m-1)$ são necessários, visto que o polinômio é do primeiro grau $(m-1)$, ou seja, uma reta e dois pontos são suficientes para defini-la.

As propriedades de interesse listadas por Shamir para seu esquema são:

1. O tamanho de cada parte não excede o tamanho do dado original;
2. Mantendo-se m fixo, outras partes podem ser adicionadas ou removidas sem afetar as demais;
3. Pode-se trocar as partes sem trocar o dado original (s), bastando um novo polinômio com o mesmo termo independente $(a_0 = s)$.

Como é empregada a aritmética modular, os coeficientes e o próprio segredo devem ser menor que o módulo. O maior número primo conhecido[1] é $2^{43.112.609} - 1$ que possui 5.389.076 bytes ($\approx 5,14$ MiB).

[1] Fonte: `http://primes.utm.edu/largest.html#largest`

Isso significa que um documento que fosse compartilhado poderia ter, no máximo, esse tamanho. Outro problema em relação ao tamanho do módulo é o custo das operações matemáticas envolvidas. Por exemplo, o cálculo da exponenciação que resulta esse número primo levou cerca de 25 min[2].

De modo a contornar tal limitação, o documento pode ser dividido em blocos, cada bloco sendo tratado como um segredo (s) do compartilhamento.

> A fim de compartilhar um segredo, escolha n valores públicos $x_1, \cdots, x_n$ diferentes entre si e um polinômio secreto f de ordem $k-1$. O segredo corresponde a $f(0)$. Aplique o algoritmo de compartilhamento de segredo, i.e. calcule os valores da função $y_i = f(x_i)$ para $i = 1, \cdots, n$. Para compartilhar um documento, divida-o em blocos de mesmo tamanho e trate cada bloco como um segredo, i.e. aplique o algoritmo mencionado acima. Para cada bloco os valores x_i são fixos enquanto o polinômio secreto varia. Quando todos os blocos de dado tiverem executado o algoritmo, os valores da função são ordenados como segue: para cada x_i, agrupe todos os valores da função $y_i = f(x_i)$ da iteração em um pacote chamado parte. Assim, para cada parte-ID x_i existe apenas uma parte. Por fim, as partes são distribuídas entre n servidores de armazenamento. A fim de reconstruir um objeto arquivado, todos os seus blocos de dados devem ser reconstruídos pela coleta de k de n partes e aplicação de uma fórmula de interpolação. (HUHNLEIN et al., 2009, tradução nossa).

Para visualizar tal algoritmo, considere o compartilhamento de um documento por n participantes. Na divisão desse documento obteve-se b blocos. Para cada b_i foram calculadas n partes $(s_{i1}, s_{i2}, \cdots, s_{in})$, representadas nas i linhas da matriz 3.2. Assim, tem-se uma matriz de ordem $b \times n$ tal que $b > n$. Entretanto, deseja-se ter n partes para o documento. Então, cada coluna c_j, para $j = 1, \cdots, n$, da matriz será uma parte $(s_{1j}, s_{2j}, \cdots, s_{bj})$ do compartilhamento do documento (matriz transposta), totalizando n partes, conforme o algoritmo descrito acima.

[2]Utilizando um processador Core 2 Duo de 2, 4GHz, sistema Mac OS X 10.6.6 e linguagem Java 1.6.0_22 (classe *BigInteger*.)

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{b1} & s_{b2} & \cdots & s_{bn} \end{pmatrix} \tag{3.2}$$

3.3 COMPARTILHAMENTO DE SEGREDO VERIFICÁVEL

Compartilhamento de segredo verificável (*Verifiable Secret Sharing* - VSS) é um protocolo que adiciona a possibilidade de um participante de um compartilhamento verificar que recebeu uma parte válida do segredo, sem que as partes dos demais participantes seja revelada. O trabalho de Chor et al. (1985) introduziu a técnica de compartilhamento de segredo verificável.

Feldman (1987) criou um esquema prático para compartilhamentos de segredo verificáveis não-interativos, usando o esquema de Shamir (1979), o qual foi incorporado em alguns protocolos, em particular nos protocolos de redistribuição de segredo (ver seção 3.5). Considerando-se os corpos Z_p e Z_r, tal que p e r são primos e $r = pq + 1$ (q é um inteiro não negativo), escolhe-se um elemento $g \in Z_r$ de ordem p (g é um gerador do grupo G), tal que $g^p \equiv 1 \pmod{r}$. Utilizando o esquema de Shamir (ver seção 3.2), calcula-se $g^{a_0} \cdots g^{a_{m-1}}$, enviando-os a todos os i participantes. Cada participante i pode verificar se s_i é uma parte válida de s comparando $g^{s_i} \equiv g^{a_0}(g^{a_1})^i \cdots (g^{a_{m-1}})^{i^{m-1}}$ (WONG; WANG; WING, 2002).

Aplicando o esquema de Feldman ao exemplo da seção 3.2, tem-se que $a_0 = 42, a_1 = 51$ e $p = 53$. Supondo $q = 2$, então $r = pq + 1 = 107$. Escolhendo-se $g = 3$, calcula-se $g^{a_0} = 3^{42} = 19 \pmod{107}$ e $g^{a_1} = 3^{51} = 12 \pmod{107}$. Cada um dos $n = 3$ participantes, então, verifica sua parte ($s_1 = 40, s_2 = 38, s_3 = 36$):

$$g^{s_1} = 3^{40} \equiv g^{a_0}(g^{a_1})^1 = 19(12)^1 = 14 \pmod{107}$$

$$g^{s_2} = 3^{38} \equiv g^{a_0}(g^{a_1})^2 = 19(12)^2 = 61 \pmod{107}$$

$$g^{s_3} = 3^{36} \equiv g^{a_0}(g^{a_1})^3 = 19(12)^3 = 90 \pmod{107}$$

Percebe-se que são equivalentes e, assim, cada participante tem a prova que sua parte é válida. Todavia, a segurança do esquema de Feldman é baseada na intratabilidade do cômputo de logaritmos discretos, ou

seja, condicionalmente seguro. Pedersen (1991) aprimorou o protocolo de Feldman para prover segurança incondicional.

A construção é similar ao esquema de Feldman, acrescido de novos valores. Escolhe-se um valor aleatório $t \in Z_p$ e calcula-se outro compartilhamento no qual t é o segredo ($b(i) = t+b_1 i+b_2 i^2+...+b_{m-1} i^{m-1}$) para computar as partes $t_i = b(i)$. Além do gerador g do esquema de Feldman, escolhe-se um gerador h (em Z_r) para as novas partes ($h^t, h^{b_1}, ..., h^{b_{m-1}}$), calculando-se $g^k h^t, g^{a_1} h^{b_1}, \cdots, g^{a_{m-1}} h^{b_{m-1}}$, enviando-os para os n participantes. Cada participante i pode verificar se s_i, t_i são partes válidas de s e t comparando $g^{s_i} h^{t_i} \equiv g^{a_0} h^{b_0} (g^{a_1} h^{b_1})^i \cdots (g^{a_{m-1}} h^{b_{m-1}})^{i^{m-1}}$ (GUPTA; GOPINATH, 2007).

Aplicando o esquema de Pedersen ao exemplo anterior e escolhendo-se $t = b_0 = 21$ e $b_1 = 23$, tem-se que $b_0 = t = 21, b_1 = 23$ e $p = 53$, resultando nas partes $b_1 = 44, b_2 = 14, b_3 = 37$. Escolhendo-se o gerador $h = 4$, calcula-se $h^{b_0} = 4^{21} = 57 \pmod{107}$ e $h^{b_1} = 4^{23} = 56 \pmod{107}$. Cada um dos $n = 3$ participantes, então, verifica suas partes (s_i e b_i):

$$g^{s_1} h^{b_1} = 3^{40} 4^{44} \equiv g^{a_0} h^{b_0} (g^{a_1} h^{b_1})^1 = 13(30)^1 = 69 \pmod{107}$$

$$g^{s_2} h^{b_2} = 3^{38} 4^{14} \equiv g^{a_0} h^{b_0} (g^{a_1} h^{b_1})^2 = 13(30)^2 = 37 \pmod{107}$$

$$g^{s_3} h^{b_3} = 3^{36} 4^{37} \equiv g^{a_0} h^{b_0} (g^{a_1} h^{b_1})^3 = 13(30)^3 = 40 \pmod{107}$$

Percebe-se que são equivalentes e, assim, cada participante tem a prova que sua parte é válida. Diferentemente do esquema de Feldman, aplicando-se este esquema tem-se uma prova com segurança incondicional. O esquema de Pedersen é aplicado ao protocolo G^2_{its}VSR que será apresentado na seção 3.5.

3.4 COMPARTILHAMENTO DE SEGREDO PROATIVO

O compartilhamento de segredo proativo (*Proactive Secret Sharing* – PSS) consiste na renovação das partes de um segredo em um compartilhamento sem que seja necessário trocá-lo. O tempo de vida do segredo é segmentado em períodos de tempo mais curtos (um mês, por exemplo) e as partes do segredo são renovadas em cada período, devendo-se descartar as partes do período anterior.

Ao invés de comprometer m partes de um segredo ao longo do tempo que a informação é armazenada confidencialmente, supondo um compartilhamento (m, n), o adversário teria de comprometer m partes do

segredo antes de uma renovação. Do mesmo modo, a destruição de $n-m$ partes do segredo teria de ser realizada em um mesmo período. Assim, a janela de tempo para um adversário móvel atacar é o período entre as renovações. Ou seja, a técnica fundamenta-se na fragmentação do período de tempo, em intervalos menores, com o intuito de inviabilizar ataques de adversários móveis.

O conceito de compartilhamento de segredo proativo foi criado com o trabalho de Herzberg, Krawczyk e Yung (1995), o qual pode suportar até $m = \frac{n}{2} - 1$ participantes corrompidos em um mesmo período de tempo. Tal esquema consiste na adição de um polinômio aleatório δ, $\delta_i(z) = \delta_{i1}z^1 + \delta_{i2}z^2 + \cdots + \delta_{im-1}z^{m-1}$, tal que $\delta(0) = 0$, a um segredo $s = f(0)$. Assim, no período t, $f^t(0) = f^{t-1}(0) + \delta(0) = s + 0 = s$. As partes do período anterior $(t-1)$ devem ser excluídas do sistema para evitar que o adversário obtenha alguma informação sobre o segredo. Enfim, as partes do segredo são renovadas sem que seja necessário revelá-lo.

Para exemplificar, parte-se do exemplo de compartilhamento de segredo visto na seção 3.2. Suponha que se queira renovar as partes $(s_1 = 40, s_2 = 38, s_3 = 36)$. Cada um dos três participantes gera um polinômio aleatório δ, tal que $\delta(0) = 0$:

$$\delta_1(z) = \delta_{11}z = 17z \text{ (mod } 53)$$

$$\delta_2(z) = \delta_{21}z = 23z \text{ (mod } 53)$$

$$\delta_3(z) = \delta_{31}z = 35z \text{ (mod } 53)$$

Após o cálculo do polinômio aleatório δ, cada participante i calcula $u_{ij} = \delta_i(j)$, para $j = 1 \cdots n$ e envia u_{ij}, sendo $i \neq j$ para os demais participantes:

$$u_{11} = \delta_1(1) = 17; u_{12} = \delta_1(2) = 34; u_{13} = \delta_1(3) = 51 \text{ (mod } 53)$$

$$u_{21} = \delta_2(1) = 23; u_{22} = \delta_2(2) = 46; u_{23} = \delta_2(3) = 16 \text{ (mod } 53)$$

$$u_{31} = \delta_3(1) = 35; u_{32} = \delta_3(2) = 17; u_{33} = \delta_3(3) = 52 \text{ (mod } 53)$$

Assim, cada participante atualiza sua parte fazendo $x_i^{t+1} = x_i^t + u_{1i} + \cdots + u_{ni}$:

$$x_1^1 = x_1^0 + u_{11} + u_{21} + u_{31} = 9 \text{ (mod } 53)$$

$$x_2^1 = x_2^0 + u_{12} + u_{22} + u_{32} = 29 \text{ (mod } 53)$$

$$x_3^1 = x_3^0 + u_{13} + u_{23} + u_{33} = 49 \text{ (mod } 53)$$

Finalmente, após esta etapa as novas partes do segredo são: $s_1 = 9, s_2 = 29, s_3 = 49$, sendo que as partes do período anterior ($s_1 = 40, s_2 = 38, s_3 = 36$) devem ser removidas pelos participantes. Aplicando-se Lagrange tem-se que:

$$f(x) = y_0 \cdot l_0 + y_1 \cdot l_1 + y_2 \cdot l_2$$

$$f(x) = 9(\frac{1}{2}x^2 - \frac{5}{2}x + 3) + 29(-x^2 + 4x - 3) + 49(\frac{1}{2}x^2 - \frac{3}{2}x + 1) \text{ (mod } 53)$$

$$f(x) = 20x + 42 \text{ (mod } 53)$$

$$f(0) = 42 = s$$

3.5 PROTOCOLOS DE REDISTRIBUIÇÃO DE SEGREDO

Os protocolos de redistribuição de segredo diferem do compartilhamento de segredo proativo uma vez que os segredos são redistribuídos – sem haver reconstrução – sendo projetados para comportar mudanças na estrutura de acesso (*access structure*), ou seja, no tamanho e configurações do grupo com o qual o segredo é compartilhado. Desmedt e Jajodia (1997) definiram o primeiro protocolo de redistribuição ao perceberem essa limitação na abordagem do compartilhamento de segredo proativo.

> Protocolos de redistribuição e esquemas de compartilhamento de segredo proativo ou permitem um participante faltoso corromper indetectavelmente a redistribuição (deixando outros participantes com partes inválidas) ou proíbem mudanças no grupo de participantes que armazenam as partes do segredo. (WONG; WANG; WING, 2002, tradução nossa).

Combinando o protocolo de Desmedt e Jajodia (1997) com o esquema de compartilhamento de segredo verificável de Feldman (1987), Wong, Wang e Wing (2002) propuseram um protocolo de redistribuição verificável (*Verifiable Secret Redistribution* – VSR) para contornar tais limitações, aplicado a sistemas de arquivamento.

Uma extensão ao VSR (*Extended Verifiable Secret Redistribution* – xVSR) foi proposta por Gupta e Gopinath (2006) na qual o requisito de que todos os participantes fossem honestos foi relaxado, requerendo que

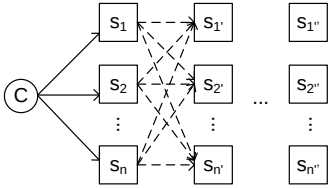

Figura 3.1: Visão Geral do Protocolo VSR

apenas a maioria seja confiável. Tal resultado foi conseguido com o uso de um mecanismo de reclamação.

Gupta e Gopinath (2007) revisaram o protocolo xVSR, denominando tal revisão de G^2_{cs}VSR. Nesse trabalho, adaptaram o protocolo revisado para utilizar o esquema de compartilhamento de segredo verificável de Pedersen (1991), propondo o *Information Theoretical Secure VSR* – G^2_{its}VSR. A troca da abordagem de VSS deu-se, pois, esse mecanismo provê sigilo teórico da informação.

Para evitar a reconstrução do segredo aplica-se uma redistribuição do mesmo. Suponha que um cliente C possui um segredo S e divide-o em n partes, armazenando-as em n servidores, de modo que com m ou mais partes seja possível reconstruir S ($m \leq n$), formando uma estrutura de acesso (m, n). Com o passar do tempo, as n partes são redistribuídas em n' partes sendo que pelo menos m' partes são necessárias para reconstruir o mesmo segredo S ($m' \leq n'$), formando a estrutura de acesso (m', n'). Essas estruturas de acesso podem ser de diferentes tamanhos, dependendo do número de servidores disponíveis quando o processo de redistribuição é executado. A qualquer momento o cliente pode reconstruir o segredo (WONG; WANG; WING, 2002). A figura 3.1 ilustra esse protocolo que é basicamente o mesmo para os três VSRs.

O processo de redistribuição utiliza um esquema de VSS para que os participantes (servidores) possam verificar a integridade e validade de suas partes do segredo.

Tais procolos são compostos por três fases: inicial, redistribuição e reconstrução. Na fase inicial o cliente C compartilha seu segredo com os servidores (n); na fase seguinte, redistribuição, as n partes geradas na fase inicial são redistribuídas, reorganizadas e reconstruídas, constituindo novas partes (n'); por fim, na fase de reconstrução o cliente requisita as partes armazenadas pelos servidores. Em todas as fases ocorrem verificações das partes pelos participantes do protocolo por meio da técnica de VSS adotada. A seguir, são descritas essas fases para o protocolo G^2_{its}VSR.

3.5.1 Preliminares

Antes de descrever as fases do procolo é necessário definir algumas informações preliminares. Essas premissas são as mesmas descritas no trabalho de Gupta e Gopinath (2007), transcritas nesta seção.

Em relação as comunicações do sistema, considera-se que:

1. os canais de comunicação garantem a entrega das mensagens de forma confiável;

2. exitem canais privados de conexão ponto-a-ponto entre cada nó;

3. existe um canal de *broadcast* do qual todo nodo participante recebe qualquer mensagem enviada por qualquer nó;

4. a identidade dos nodos não pode ser forjada.

O protocolo descrito utiliza o compartilhamento de segredo de Shamir (1979), sendo k e $t \in \mathbb{Z}_p$.

As estruturas de acesso denominadas por (m, n) significam que um segredo é compartilhado por n servidores de modo que pelo menos $m \leq n$ são necessários e suficientes para reconstrução do segredo. Além disso, $n \leq 2m-1$, de forma que o segredo possa ser reconstruído sempre por uma maioria honesta. Ainda, denota-se um conjunto de servidores admissíveis por $\mathcal{B}$ (ver seção 3.2).

Sejam p e r primos grandes e $r = pq + 1$, sendo q um inteiro não negativo. $\mathbb{Z}_p$ e $\mathbb{Z}_r$ são corpos finitos com a aritmética módulo p e módulo r, respectivamente. Os valores g e $h \in Z_r$ são geradores do corpo finito $\mathbb{Z}_r$ tal que $g^p \equiv 1 \bmod r$ e $h^p \equiv 1 \bmod r$.

Seja um algoritmo $\mathcal{A}(x, I, \mathcal{Z})$ no qual x é um inteiro e I um conjunto com a identidade dos servidores, com y elementos, tal que $x < y$. O resultado $\mathcal{Z}$ é uma sequência de subconjuntos de I cada qual contendo exatamente x membros. Existem "y escolhe x"membros na sequência cada qual contendo uma combinação de x servidores do conjunto I.

3.5.2 Inicial

Seja C um cliente que distribui um segredo k para n servidores com uma estrutura de acesso (m, n). Essa distribuição ocorre da seguinte forma:

1. C escolhe um número t aleatório de $\mathbb{Z}_p$ e dois polinômios $a(i) = k + a_1 i + \cdots + a_{m-1} i^{m-1}$ e $b(i) = t + b_1 i + \cdots + b_{m-1} i^{m-1}$ para calcular as partes $s_i = a(i)$ e $t_i = b(i)$, de k e t, e envia o par de partes (s_i, t_i) para o servidor i;

2. C usa os geradores g e h para calcular $g^k, g^{a_1}, \cdots, g^{a_{m-1}}$ e $h^t, h^{b_1}, \cdots, h^{b_{m-1}}$ e para calcular as testemunhas $g^k h^t$, $g^{a_1} h^{b_1}$, $\cdots$, $g^{a_{m-1}} h^{b_{m-1}}$, enviando-as para todos os servidores via *broadcast*;

3. Cada servidor i verifica se:

$$g^{s_i} h^{t_i} \equiv g^k h^t \prod_{l=1}^{m-1} (g^{a_l} h^{b_l})^{i^l} \tag{3.3}$$

- Se a verificação proceder o servidor i mantém as partes (s_i, t_i);
- Caso a verificação falhe o servidor i registra uma reclamação do cliente e inicia-se o sub-protocolo de reclamação. Esse sub-protocolo tem por objetivo verificar quem está sendo desonesto, se é o cliente ou o servidor:

(a) o cliente envia pelo canal de *broadcast* as partes $(\hat{s}_i, \hat{t}_i)$ enviadas ao servidor i;

(b) os servidores verificam as partes reveladas pela equação 3.3

- Se a verificação proceder os demais servidores marcam o servidor i como desonesto;
- Caso a verificação falhe o cliente é marcado como desonesto e o protocolo é abortado.

4. A maioria honesta dos servidores recebeu o par (s_i, t_i) correto e o armazena junto com a testemunha $g^k h^t$.

Gupta e Gopinath (2007) denotaram as partes (s_i, t_i) como $(\hat{s}_i, \hat{t}_i)$ pois elas podem ser diferentes das partes do passo anterior, visto que o cliente pode ter modificado as partes (s_i, t_i), caso seja desonesto.

3.5.3 Redistribuição

Redistribui-se o par (k, t), composto pelas partes (s_i, t_i), da estrutura de acesso (m, n) para (m', n'). Os servidores da primeira estrutura são os emissores e da segunda, os receptores. Os passos dessa etapa de redistribuição são:

1. Cada emissor i escolhe dois polinômios $a'_i(j) = s_i + a'_{i1} j + \cdots + a'_{i(m'-1)} j^{m'-1}$ e $b'_i(j) = t_i + b'_{i1} j + \cdots + b'_{i(m'-1)} j^{m'-1}$ para calcular as partes $\hat{s}_{ij} = a'_i(j)$ e $\hat{t}_{ij} = b'_i(j)$, de s_i e t_i, e envia o par de partes $(\hat{s}_{ij}, \hat{t}_{ij})$ para o servidor j;

2. Cada emissor i usa os geradores g e h para calcular $g^{s_i}, g^{a'_1}, \cdots, g^{a'_i(m'-1)}$ e $h^{t_i}, h^{b'_i 1}, \cdots, h^{b'_i(m'-1)}$ e para calcular as testemunhas $g^{s_i} h^{t_i}$, $g^{a'_{i1}} h^{b'_{i1}}$, $\cdots$, $g^{a'_{i(m'-1)}} h^{b'_{i(m'-1)}}$, enviando-as para todos os receptores via *broadcast*;

3. Cada receptor j verifica se:

$$\forall i, g^{\hat{s}_{ij}} h^{\hat{t}_{ij}} \equiv g^{s_i} h^{t_i} \prod_{l=1}^{m'-1} (g^{a'_{il}} h^{b'_{il}})^{j^l} \tag{3.4}$$

   - Se a verificação proceder o servidor j mantém as subpartes $(\hat{s}_{ij}, \hat{t}_{ij})$;
   - Caso a verificação falhe o servidor j registra uma reclamação do servidor i e inicia-se o sub-protocolo de reclamação:

   (a) o servidor i envia pelo canal de *broadcast* as subpartes $(\hat{\hat{s}}_{ij}, \hat{\hat{t}}_{ij})$ enviadas ao servidor j;

   (b) os servidores verificam as partes reveladas pela equação 3.4
      - Se a verificação proceder os demais servidores marcam o servidor j como desonesto;
      - Caso a verificação falhe o servidor i é marcado como desonesto.

4. Cada receptor utiliza o algoritmo $\mathcal{A}$ para formar uma sequência $\mathcal{B}_u$ com m-subconjuntos de U servidores (número de emissores não marcados), onde $u = 1, \cdots, M$, $1 \leq M \leq N$ e N sendo "U escolhe m";

5. Cada emissor envia por *broadcast* a testemunha $g^k h^t$. Pela maioria honesta, os receptores terão a testemunha correta para realizar a verificação no passo seguinte;

6. Os receptores verificam em sucessivos $\mathcal{B}_u$ a validade das partes por:

$$g^k h^t \equiv \prod_i (g^{s_i} h^{t_i})^{b_i} \text{ onde } b_i = \prod_{l \in \mathcal{B}_u, l \neq i} \frac{l}{l-i}$$

7. Cada receptor j calcula seu novo par de partes, usando as subpartes do conjunto de emissores B_u, pela fórmula:

$$s'_j = \sum_{i \in \mathcal{B}_u} b_i \hat{s}_{ij} \text{ e } t'_j = \sum_{i \in B} b_i \hat{t}_{ij} \text{ onde } b_i = \prod_{l \in \mathcal{B}_u, l \neq i} \frac{l}{l-i}$$

   e armazena o par (s'_j, t'_j) e a testemunha $g^k h^t$.

3.5.4 Reconstrução

Seja C um cliente, cujo segredo k foi (re)distribuído e que deseja obtê-lo novamente. Os passos para reconstrução do segredo k, a partir da estrutura de acesso (m, n), são:

1. C requisita as partes do segredo. Cada servidor i envia seu par (s_i, t_i) para C;

2. C obtém a testemunha $g^k h^t$ pela maioria dos servidores;

3. C utiliza o algoritmo $\mathcal{A}$ para formar uma sequência $\mathcal{B}_u$ com m-subconjuntos de n servidores, onde $u = 1, \cdots, M$, $M \leq N$ e N sendo "n escolhe m";

4. C verifica em sucessivos $\mathcal{B}_u$ a validade das partes por:

$$g^k h^t \equiv \prod_i (g^{s_i} h^{t_i})^{b_i} \text{ onde } b_i = \prod_{l \in \mathcal{B}, l \neq i} \frac{l}{l - i}$$

5. C reconstrói o segredo k, usando o primeiro $\mathcal{B}_u$ para o qual o teste aprovar, pela equação:

$$k = \sum_i s_i b_i \text{ onde } b_i = \prod_{l \in \mathcal{B}_u, l \neq i} \frac{l}{l - i}$$

Para exemplificar, parte-se do exemplo de compartilhamento de segredo visto na seção 3.2, utilizando as mesmas variáveis e configurações. Adicionalmente, suponha que se escolheu o gerador $g = 3$ em $\mathbb{Z}_r$, tal que $r = pq + 1 = 53 \cdot 2 + 1 = 107$ (ver seção 3.3). Neste exemplo é mostrada a execução do protocolo xVSR.

Na fase inicial o cliente gera as partes $s_1 = 40$, $s_2 = 38$ e $s_3 = 36$. Utilizando o gerador $g = 3$ o cliente calcula as testemunhas $g^k = 19$ e $g^{a_1} = 12$. As partes e testemunhas são enviadas para os respectivos servidores por conexões privadas. Cada servidor verifica sua parte e, em caso de sucesso, armazena sua parte respectiva:

$$g^{s_1} = 14 \equiv g^k (g^{a_1})^1 = 19 \cdot 12^1 = 14$$
$$g^{s_2} = 61 \equiv g^k (g^{a_1})^2 = 19 \cdot 12^2 = 61$$
$$g^{s_3} = 90 \equiv g^k (g^{a_1})^3 = 19 \cdot 12^3 = 90$$

Na fase de redistribuição cada servidor i compartilha sua parte por $(m' = 2, n' = 3)$, gerando $n \cdot n' = 9$ partes temporárias ($[51, 9, 20]$, $[49, 7, 18]$ e $[47, 5, 16]$), considerando $a_1 = 11$. Utilizando o gerador

$g = 3$, cada emissor calcula as testemunhas $g^{a_1} = 62$. Cada receptor j de n', verifica suas partes temporárias e as partes iniciais:

$$
\begin{aligned}
g^{\hat{s}_{11}} = 12 \equiv g^{s_1}(g^{a_1})^1 = 14 \cdot (62)^1 = 12 \\
g^{\hat{s}_{12}} = 102 \equiv g^{s_1}(g^{a_1})^2 = 14 \cdot (62)^2 = 102 \\
g^{\hat{s}_{13}} = 11 \equiv g^{s_1}(g^{a_1})^3 = 14 \cdot (62)^3 = 11 \\
g^{\hat{s}_{21}} = 37 \equiv g^{s_2}(g^{a_1})^1 = 61 \cdot (62)^1 = 37 \\
g^{\hat{s}_{22}} = 47 \equiv g^{s_2}(g^{a_1})^2 = 61 \cdot (62)^2 = 47 \\
g^{\hat{s}_{23}} = 25 \equiv g^{s_2}(g^{a_1})^3 = 61 \cdot (62)^3 = 25 \\
g^{\hat{s}_{31}} = 16 \equiv g^{s_3}(g^{a_1})^1 = 90 \cdot (62)^1 = 16 \\
g^{\hat{s}_{32}} = 29 \equiv g^{s_3}(g^{a_1})^2 = 90 \cdot (62)^2 = 29 \\
g^{\hat{s}_{33}} = 86 \equiv g^{s_3}(g^{a_1})^3 = 90 \cdot (62)^3 = 86 \\
g^k = 19 \equiv (g^{s_1})^3(g^{s_2})^{-3}(g^{s_3})^1 = 19
\end{aligned}
$$

Ainda na redistribuição, cada servidor i mantém a parte temporária s_{ij} na qual $j = i$, enviando as demais para os respectivos j, resultando nas partes $[51, 49, 47]$, $[9, 7, 5]$ e $[20, 18, 16]$. Por fim, cada receptor reconstrói por Lagrange suas partes temporárias, resultando nas novas partes $[0, 11, 22]$.

Finalmente, na fase de reconstrução o cliente recebe as partes $[0, 11, 22]$, verifica-as, e reconstrói o segredo por Lagrange:

$$g^k = 19 \equiv (g^{s_1})^3(g^{s_2})^{-3}(g^{s_3})^1 = 19$$

$$k = s_1b_1 + s_2b_2 + s_3b_3 = 0 \cdot 3 + 11 \cdot -3 + 22 \cdot 1 = 42$$

Percebe-se que as partes iniciais $[40, 38, 36]$, após a redistribuição são alteradas para $[0, 11, 22]$, entretanto, reconstroem o mesmo segredo $k = 42$.

3.6 SINTAXE DO REGISTRO DE EVIDÊNCIA

Conforme visto na seção 2.6.1, a preservação da assinatura digital provê a autenticidade de documentos eletrônicos. Verificou-se também, na seção 2.6.2, que carimbos do tempo são uma possível forma de se preservar assinaturas digitais por longo prazo.

Formatos de assinatura como CAdES (ETSI, 2009a), XAdES (ETSI, 2009c) e PAdES (ETSI, 2009b) são ineficientes para grandes volumes de documentos uma vez que é necessário pelo menos um carimbo

do tempo para cada documento arquivado. Além disso, existindo mais de uma cópia do documento eletrônico, uma situação comum dadas as características do documento eletrônico, torna-se complexo gerenciar os carimbos do tempo para as cópias, por exemplo, pode haver uma cópia com e outra sem carimbo ou com carimbos distintos. Um meio de otimizar essa situação é com o emprego da sintaxe de registro de evidência.

A sintaxe de registro de evidência (*Evidence Record Syntax* – ERS) especifica uma sintaxe para um registro de evidência (*Evidence Record*), que contém um conjunto de carimbos do tempo de arquivamento e alguns dados adicionais, e especifica processos para geração e verificação desses registros (BRANDNER; PORDESCH; GONDROM, 2007). Os dados são armazenados em uma estrutura de dados normalmente descrita em *Abstract Syntax Notation One* (ASN.1), que é o registro de evidência.

O registro de evidência é baseado na árvore de Merkle (1980) que consiste em uma árvore na qual os nós são resumos criptográficos. Os resumos são ordenados de forma binária ascendente, concatenados aos pares, dos quais se extrai um novo resumo criptográfico até que reste um único resumo no nó raiz. Caso o número de resumos seja ímpar, Merkle propôs que ele seja utilizado quando houver outro número ímpar de nós. A figura 3.2 ilustra tais configurações, supondo a ordem binária $h_1 < h_2 < h_3 < h_4$, $h_{12} < h_{34}$ e $h_{12} < h_3$.

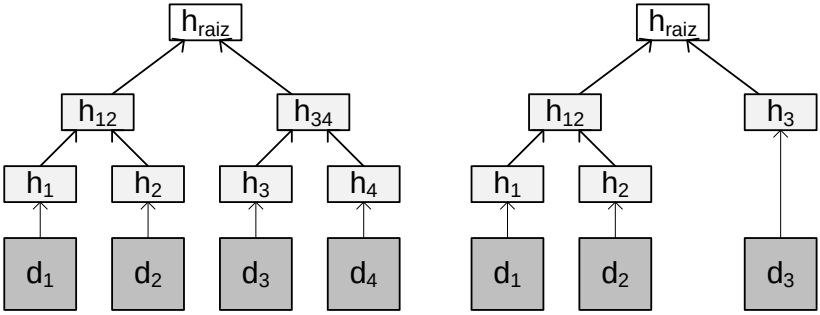

Figura 3.2: Árvores de Merkle balanceada (esquerda) e desbalanceada (direita).

No caso do ERS é criada uma árvore de Merkle a partir do resumo criptográfico dos objetos que se deseja preservar, calculando-os até se obter um único resumo na raiz. Esse resumo criptográfico recebe, então, um carimbo do tempo. Para se provar que um objeto mantém-se autêntico, calcula-se o resumo criptográfico do objeto e, utilizando os resumos dos nós irmão, concatena-se e calcula-se os resumos criptográficos até chegar à raiz, na qual, validando-se o carimbo do tempo, valida-se também o objeto associado. A figura 3.3 ilustra os resumos necessários (hachurados), a chamada árvore reduzida, na verificação de pertinência à árvore do objeto d_3. Percebe-se que os resumos necessários são h_3, h_4 e h_{12} pois os demais são calculáveis a partir desses resumos.

Entretanto, a obsolescência também ocorre com o ERS, do mesmo

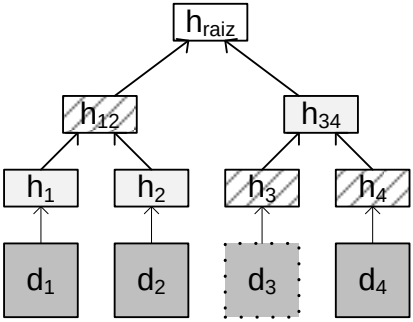

Figura 3.3: Árvore reduzida para o objeto d_3.

modo que as assinaturas digitais (ver seção 2.6.1), sendo preciso renovar as tecnologias envolvidas para garantir a manutenção das propriedades de segurança por longo prazo. São possíveis dois tipos de renovação: a renovação do carimbo do tempo (*Time-Stamp Renewal*) e a renovação da árvore de resumo criptográfico (*Hash-Tree Renewal*).

A primeira ocorre se os algoritmos ou parâmetros criptográficos empregados tornarem-se fracos ou o certificado da carimbadora ou de alguma autoridade certificadora do caminho de certificação expirar ou for revogado. Nestes casos, antes que o evento aconteça, deve-se recarimbar o carimbo do tempo da raiz da árvore.

O segundo caso ocorre quando o algoritmo de resumo criptográfico utilizado na montagem da árvore torna-se inseguro. Antes que tal evento ocorra, deve-se renovar toda a árvore, a partir da árvore antiga, obtendo-se o resumo criptográfico dos objetos envolvidos e, finalmente, construindo-se uma nova árvore a qual um novo carimbo do tempo é adicionado à raiz.

3.7 SISTEMAS DE ARQUIVAMENTO

Diversas propostas de sistemas de arquivamento com as mais variadas configurações e propostas são encontrados na literatura. Storer, Greenan e Miller (2006) realizaram um levantamento das principais propostas de sistemas de arquivamento e o refinaram mais tarde (STORER et al., 2009). Reproduz-se, na tabela 3.1, as informações de Storer et al. (2009), adaptando-as ao escopo deste trabalho.

Observa-se inicialmente que nenhum dos sistemas listados na tabela 3.1 emprega esquemas para manutenção da autenticidade. Assim, essas propostas estão impossibilitadas de receber informações digitalmente assinadas, visto que as assinaturas digitais precisam de constante renovação, por exemplo, por meio de carimbos do tempo (HABER; STORNETTA, 1991). Parte-se, então, para a análise das técnicas de confidencialidade adotadas.

Sistema	Confidencialidade	Integridade	Blocos para Comprometimento
FreeNet	cifragem	resumo criptográfico	1
OceanStore	cifragem	versionamento	m (de n)
FarSite	cifragem	árvores de Merkle	1
Publius	cifragem	baseado em recuperação	m (de n)
SNAD/Plutus	cifragem	resumo criptográfico	1
SafeStore	cifragem	resumo criptográfico	m (de n)
GridSharing	segredo compartilhado	replicação	1
PASIS	segredo compartilhado	agentes de reparação/auditoria	m (de n)
CleverSafe	IDA	resumo criptográfico	m (de n)
Glacier	cifragem do usuário	assinaturas	–
Venti	–	recuperação	–
LOCKSS	–	verificação baseada em voto	–
POTSHARDS	segredo compartilhado	assinaturas algébricas	$O(R^{m-1})$

Tabela 3.1: Sistemas de arquivamento.

Os sistemas Venti e LOCKSS não possuem funcionalidade de confidencialidade. FreeNet, OceanStore, FarSite, Publius, SNAD/Plutus, SafeStore e Glacier empregam a cifragem como meio de prover confidencialidade. CleverSafe utiliza algoritmos de dispersão. Os demais, GridSharing, PASIS e POTSHARDS, utilizam segredo compartilhado como técnica de sigilo.

Em relação à manutenção da integridade, excetuando FarSite – que utiliza árvores de Merkle, todos os sistemas utilizam métodos simples de manutenção dessa propriedade.

O projeto ArchiSig[3] (BRANDNER; PORDESCH, 2002) aplicou as árvores de Merkle para manutenção da integridade e autenticidade, originando o grupo de trabalho Long-Term Archiving & Notary Service[4] (LTANS) do IETF e, posteriormente, o ERS.

O projeto ArchiSafe[5] (ZIMMER; LANGKABEL; HENTRICH, 2008) foi conceitualmente construído com base nos resultados do projeto ArchiSig, consistindo em um arcabouço (*framework*) legal para o gerenciamento e arquivamento por longo prazo de documentos eletronicamente assinados. Os formatos padronizados de arquivos eletrônicos aceitos são TXT, PDF (e PDF/A), TIFF e XML. Um container XML encapsula os objetos arquivados (binários são convertidos em base 64), metadados e, opcionalmente, um bloco para assinatura e carimbo do tempo. A integridade e autenticidade dos objetos arquivados é realizada por meio do ERS. Entretanto o projeto não prevê a confidencialidade dos objetos arquivados, exceto pelo controle de acesso (ver seção 2.4.1).

[3] http://www.archisig.de
[4] http://tools.ietf.org/wg/ltans
[5] http://www.archisafe.de

Baseado nos resultados dos projetos ArchiSig e ArchiSafe, em conjunto com o modelo de referência *Open Archival Information System* (OAIS) e a Sintaxe do Registro de Evidência (ERS), o Escritório Federal para Segurança da Informação (*Federal Office for Information Security*) desenvolveu uma diretiva técnica que regulamenta o arquivamento confiável por longo prazo para agências governamentais da Alemanha, provendo integridade e autenticidade aos dados arquivados.

O modelo de referência Open Archival Information System (OAIS) (CCSDS, 2002), norma da *International Organization for Standardization* (ISO) 14721:2003 (ISO, 2003), é um arcabouço conceitual que define uma terminologia e um conjunto de conceitos, em termos análogos aqueles já usados em disciplinas e organizações de preservação da informação, visando elucidar as principais atividades e identificando áreas potenciais para o desenvolvimento de padrões relacionados, sem especificar uma forma de implementação em particular, mas tendo requisitos mínimos para estar em conformidade. Alguns projetos de sistemas de arquivamento já implementados baseiam-se no OAIS[6].

Huhnlein et al. (2009) estenderam essa arquitetura, combinando os diferentes requisitos envolvidos, para prover também confidencialidade e disponibilidade. Na seção seguinte, traz-se uma visão geral dessa arquitetura.

3.8 ARQUITETURA DE REFERÊNCIA

O trabalho de Huhnlein et al. (2009) propôs uma arquitetura de referência para o arquivamento de longo prazo empregando o Modelo de Referência OAIS, a sintaxe de registro de evidência (ERS) e o compartilhamento de segredo, sendo um trabalho importante devido à sua tentativa de lidar com a autenticidade e a confidencialidade de documentos na mesma infraestrutura, com base nestas tecnologias. Outros trabalhos de sistemas de arquivamento, como visto na seção 3.7, excluíam uma ou outra propriedade, dada a dificuldade em se lidar com ambas simultaneamente.

Essa arquitetura suporta operações para submissão, recuperação, requisição de evidência, requisição de dados complementares (como metadados) e exclusão de objetos arquivados. Um identificador (ID) dos objetos arquivados, chamado de *Archive Token*, é retornado ao cliente após a submissão de um arquivo. O *Archive Token* é utilizado para identificar o objeto arquivado nas operações requisitadas. A figura 3.4 ilustra os módulos e operações da infraestrutura proposta.

O módulo *Archive Gateway* recebe as requisições das aplicações e encapsula os objetos a serem arquivados em um pacote chamado *XML Archival Information Package (XAIP)* – se o objeto não estiver nesse pacote, controla os processos e formatos por meio de *XML Schemas* padroniza-

[6] http://www.oclc.org/research/activities/past/rlg/oaisactivities.htm

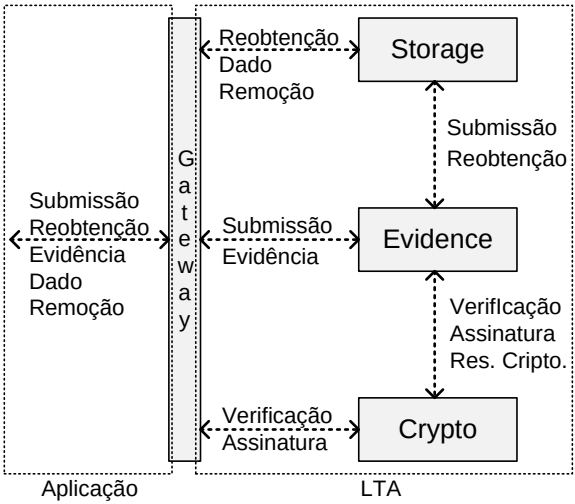

Figura 3.4: Visão geral dos módulos e operações da arquitetura de referência.

dos e realiza o controle de acesso. No caso de o pacote de arquivamento já estar assinado na camada de aplicação, o módulo Crypto é invocado a fim de verificar as assinaturas encontradas. Por fim, este módulo requisita os serviços de evidência do módulo *Evidence* e armazena o pacote de arquivamento no módulo *Storage*.

O módulo *Evidence* gera resumos criptográficos (hash) de todos os documentos, agrupado-os em árvores de resumo criptográfico (MERKLE, 1980). Um carimbo do tempo é gerado para o resumo criptográfico da raiz da árvore, consolidando a validade de todos os documentos envolvidos. Antes dos algoritmos ou parâmetros criptográficos tornarem-se fracos ou comprometidos o carimbo do tempo é renovado. Quando requisitado, o módulo utiliza essas árvores para gerar um registro de evidência, em conformidade com o padrão ERS (BRANDNER; PORDESCH; GONDROM, 2007).

O módulo *Crypto* suporta as operações criptográficas necessárias a infraestrutura, como funções de resumo criptográfico, carimbos do tempo e, opcionalmente, geração e verificação de assinaturas. A interface do módulo *Crypto* deve ser compatível com interfaces padrões, garantindo a interoperabilidade entre diferentes módulos *Crypto*.

O módulo *Storage*, além dos serviços de busca, deleção e reprodução bit-a-bit, deve prover uma arquitetura distribuída que garanta uma confidencialidade de limiar (k, n). O esquema de compartilhamento de segredo de Shamir (1979) foi adotado.

Seguindo o padrão ERS, a infraestrutura executa continuamente os processos de renovação do carimbo do tempo e da árvore de resumo criptográfico. O primeiro, conforme descrito na seção 3.6, consiste em re-

carimbar a raiz das árvores antes que aconteçam eventos que invalidem o carimbo – como expiração e revogação de certificados e obsolescência de algoritmos ou parâmetros criptográficos. O segundo caso ocorre quando o algoritmo de resumo criptográfico é considerado inseguro, devendo-se substituí-lo por um algoritmo seguro.

O mecanismo de renovação das partes (*Share Renewal*) que é descrito consiste na renovação das partes sempre que um servidor for adicionado ou removido do sistema. Quando um servidor é adicionado não é preciso alterar as partes existentes. Entretanto, quando um servidor é removido deve-se recalcular as partes para invalidar a parte removida. Tal renovação tem impacto no gerenciamento de carimbos do tempo (*Archive Timestamp Management*), dependendo da configuração adotada. Há dois modos: *Inside* e *Outside Shared Mode*.

No *Inside Shared Mode* o resumo criptográfico é calculado sobre o arquivo e somente no módulo *eSafe* é aplicado o compartilhamento de segredo. O tamanho da árvore depende do número de objetos armazenados. Considerando um limiar (k, n), onde n é total de partes e k é o número mínimo necessário para a reconstrução do segredo, são acessadas k partes para a renovação da árvore de resumo criptográfico.

No *Outside Shared Mode* o mecanismo de compartilhamento de segredo é realizado na camada de aplicação e o sistema enxerga as partes como vários documentos, sendo o resumo criptográfico calculado por cada parte. O tamanho da árvore depende do número de partes. Para a renovação da árvore de resumo criptográfico é necessário acessar todos as n partes.

3.9 CONCLUSÃO

Nesta capítulo revisou-se os trabalhos relacionados de alguma forma com a presente dissertação, disponíveis na literatura. Descreveu-se as principais características, contribuições e limitações de cada um deles.

Os seguintes trabalhos relacionados as técnicas de confidencialidade foram descritos: compartilhamento de segredo, compartilhamento de segredo verificável, compartilhamento de segredo proativo e protocolos para redistribuição de segredo. Elaborou-se um exemplo no qual cada uma das técnicas foi aplicada, clarificando e comprovando tais propostas.

Descreveu-se a sintaxe de registro de evidência como uma forma de preservação da autenticidade da informação. Como visto, tal proposta emprega árvores de resumo criptográfico e carimbos do tempo apenas na raiz dessas árvores, de modo a tornar a manutenção da autenticidade mais eficiente.

Mostrou-se alguns sistemas de arquivamento e constatou-se que somente uma proposta atendia os requisitos de confidencialidade e autenticidade com as características desejáveis. Assim, elaborou-se uma visão geral dessa proposta – a arquitetura de referência. Nessa arquitetura verificou-se que tal proposta foi descrita sucintamente, dificultando o en-

tendimento da interação entre os componentes da infraestrutura, e quando e qual módulo realizava a operação.

No capítulo 4, as técnicas de confidencialidade vistas no capítulo 2 e trabalhos relacionados descritos neste capítulo, são avaliados em relação ao longo prazo. Igualmente, verifica-se a resistência dessas propostas em relação aos modelos de adversários, também vistos no capítulo anterior. Por fim, verifica-se a arquitetura de referência visualizada neste capítulo com base nas análises realizadas.

4 TÉCNICAS CRIPTOGRÁFICAS A LONGO PRAZO

4.1 INTRODUÇÃO

Neste capítulo, analisa-se as técnicas de confidencialidade vistas na seção 2.4 quanto à manutenção de suas propriedades com o tempo, e os trabalhos relacionados mostrados no capítulo 3, a partir das definições de segurança vistas na seção 2.2. Essa analise consta na seção 4.2.

Igualmente, verifica-se a resistência dessas técnicas aos modelos de adversários descritos na seção 2.5 e possíveis soluções ou alternativas na seção 4.3. Ambas análises baseiam-se nos trabalhos e discussões encontrados na literatura científica.

A partir dessas análises, na seção 4.4, verifica-se a arquitetura de referência apresentada na seção 3.8, buscando-se avaliá-la com relação ao que ela se propõe.

Por fim, faz-se as conclusões deste capítulo na seção 4.5.

4.2 ANÁLISE

O levantamento das técnicas que promovem a confidencialidade da informação realizado na seção 2.4 permite dividir as técnicas em dois grandes grupos: técnicas primordiais e técnicas modernas.

Classificou-se como primordiais aquelas técnicas mais simples que podem ser utilizadas mesmo sem um computador[1]. Tais técnicas incluem o controle de acesso, a segurança por obscuridade e a esteganografia.

Considera-se modernas as técnicas que exploram o processamento por computador, devido à complexidade e quantidade de cálculos exigidos. Estão inclusas nesta categoria a dispersão da informação, as técnicas temporais, o ciframento e o compartilhamento de segredo.

4.2.1 Técnicas Primordiais

A técnica de controle de acesso utiliza mecanismos para segregar o acesso a um recurso ou informação. Esta técnica é largamente utilizada em diversos níveis, como organizacional, físico e lógico. Quando implantada corretamente é eficiente ao que se propõe. Entretanto, vários fatores podem burlar a proteção imposta pelo controle de acesso e, consequentemente, expor a informação sigilosa.

Fatores como corrupção de agentes de controle, falhas em sistemas e softwares, senhas óbvias ou simples de se buscar por força bruta, entre outros, tornam o controle de acesso simplório em relação ao sigilo

[1] Outras técnicas, como o ciframento, podem ser executadas sem um computador. Entretanto, se executadas neste, promovem um nível de segurança superior.

da informação. Vazamento de informações sigilosas são constantemente noticiados na mídia, em grande parte devido ao uso deste controle, unicamente. É uma técnica que deve ser usada em conjunto com outras, com fins de complementação.

Ataques de *cross-site scripting* (XSS) são definidos por Di Lucca et al. (2005, tradução nossa) como "vulnerabilidade de uma aplicação Web causada por uma falha da aplicação em verificar as entradas do usuário antes de retorná-la ao navegador do cliente". Ainda de acordo com o referido autor, "Sem uma adequada validação, entradas de usuário podem conter códigos de script maliciosos [...] causando assim uma exploração de segurança", podendo inclusive "contornar perímetros de defesa". Ou seja, tais tipos de ataque podem burlar o controle de acesso de uma página web obtendo, assim, acesso aos dados internos do sistema que em tese não deveriam ser externalizados.

Em relação as técnicas de segurança por obscuridade, os princípios de Kerckhoffs demonstram a fragilidade deste grupo de técnicas. Algoritmos públicos elaborados por toda uma comunidade tendem a ser melhores e mais seguros quando comparados a esforços privados. Mesmo algoritmos privados podem ser seguros, todavia sua segurança não deveria depender da manutenção do algoritmo em sigilo.

Técnicas de engenharia reversa, depuração e conjuntos de entrada e saída, por exemplo, podem revelar informações sobre os mecanismos do algoritmo ou mesmo quebrá-lo. Uma vez descoberto como o algoritmo opera pode-se reverter as operações e, assim, obter a informação sigilosa.

Um exemplo dessa fragilidade são os seriais empregados na distribuição e controle de softwares proprietários. Com a descoberta da fórmula de geração dos seriais, seja por engenharia reversa, depuração ou deduções a partir de conjuntos de entrada e saída, atacantes podem criar seriais válidos e burlar o controle imposto.

Já em relação as técnicas de esteganografia tem-se duas categorias: as técnicas antigas e as modernas. Nas primeiras, do mesmo modo que nas técnicas de obscuridade, a esteganografia depende do sigilo do algoritmo utilizado para obter o sigilo da mensagem, por ocultação. As técnicas modernas, por exemplo, que dependam do conhecimento de uma chave, já possuem uma segurança aprimorada, mas condicional. De acordo com Anderson e Petitcolas (1998, tradução nossa), "Ainda não temos uma teoria de esteganografia comparável"em relação a teoria da informação de Shannon (1948), ou seja, que promova segurança incondicional.

4.2.2 Técnicas Modernas

As técnicas de dispersão da informação não foram criadas com a finalidade de confidencialidade. De acordo com Rabin (1989), existe alguma revelação da informação F se alguma das n partes for obtida por um adversário.

Técnicas temporais promovem a confidencialidade da informação

de acordo com o mecanismo interno utilizado. A autoridade certificadora temporal (CUSTóDIO et al., 2007) baseia-se em criptografia assimétrica. Já o mecanismo proposto por (RIVEST; SHAMIR; WAGNER, 1996) emprega a resolução de quebra-cabeças para liberação da informação sigilosa.

Todavia, apesar de constituírem mecanismos diferentes, tais técnicas possuem um ponto em comum: a temporalidade. Configura-se um tempo no qual a informação será revelada e, consequentemente, limita-se a aplicação deste grupo para longo prazo dada a imprevisibilidade temporal. A abertura de um testamento ilustra essa questão.

As técnicas analisadas até aqui ou promovem uma fraca confidencialidade ou não a promovem. O ciframento e o compartilhamento de segredo e criptografia de limiar permitem, de fato, a confidencialidade da informação. Entretanto, precisa-se analisar tais técnicas em relação ao tempo e a adequabilidade destas para longo prazo.

Os algoritmos de ciframento são computacionalmente seguros, ou seja, sua segurança está baseada na resolução de problemas complexos, como o logaritmo discreto e a fatoração de números em primos, o que seria impraticável mesmo que se utilizasse todos os computadores existentes. Todavia, o poder computacional continua a crescer, seja por evoluções na área da computação ou eletrônica, ou ainda por expansão da quantidade de máquinas e dispositivos existentes. Além disso, tem-se pesquisas na área de criptoanálise com o intuito de encontrar vulnerabilidades nos algoritmos. É comum, em longo prazo, esperar-se encontrar vulnerabilidades em qualquer algoritmo criptográfico, mesmo que parcialmente. Qualquer avanço é significativo, desde que se reduza a complexidade esperada para a quebra do algoritmo. Uma ilustração dessa disputa entre criptólogos e criptoanalistas é o contínuo aumento do tamanho das chaves dos algoritmos, a fim de manter a complexidade do problema e, assim, a segurança do algoritmo.

Embora seja verdade que as técnicas de ciframento são desenvolvidas para serem seguras por um longo período, sua adequabilidade degrada-se com o tempo. Longo prazo significa períodos de tempo mais abrangentes que o tempo de vida previsto para as técnicas criptográficas, por exemplo, indefinidamente.

Para exemplificar a necessidade de sigilo por longo prazo lista-se alguns desses tipo de documentos. Tem-se arquivos secretos militares e de governo, que necessitam de sigilo contra inimigos da nação. Mesmo vazamento de documentos históricos pode ser danoso, pois adversários podem aprender segredos sobre a nação. Outros exemplos são registros médicos, processos judiciais, segredos industriais, entre outros.

Os algoritmos de cifragem cumprem bem seu papel até médios períodos de tempo. Entretanto, ao se projetar a necessidade de preservação da confidencialidade de uma informação para um período de tempo maior, ou indeterminado, esbarra-se nos problemas descritos anteriormente. Seria preciso renovar o ciframento para manter a informação sigilosa.

A renovação do ciframento dessas informações pode acontecer de duas formas: a primeira com a decifragem e posterior cifragem, com novos algoritmos ou parâmetros; ou uma cifragem em cima da cifra anterior. Ambas as abordagens são problemáticas. Na primeira há uma clara exposição da informação confidencial em algum local do sistema e, no caso de um adversário conseguir acesso, terá a informação sigilosa. Esse problema é intuitivamente percebido, mas também é apontado na literatura. Wong, Wang e Wing (2002) apontam que existe um servidor central para recuperação (renovação da criptografia) que pode ser comprometido por um adversário que, assim, obtém a informação confidencial. Na segunda existe a necessidade do gerenciamento de chaves, pois será preciso guardar as chaves antigas para decifrar os documentos, além da questão de obsolescência. Em ambos os casos, ainda se tem todo um processamento para essa renovação que será crescente à medida que novos documentos forem adicionados ao sistema.

A questão da obsolescência diz respeito à disponibilidade desse algoritmo no futuro. Suponha que houve pelo menos uma renovação de algoritmo e que foram preservadas as informações necessárias para decifrar a informação – qual o algoritmo, parâmetros e chaves utilizados. O sistema operacional pode não ter mais esse algoritmo disponível, ter havido uma evolução dos sistemas e quebrado a compatibilidade com esse algoritmo ou não ser possível portá-lo a um novo sistema. Enfim, a obsolescência é um tema complexo e preocupante, não só para algoritmos, como para toda a computação, bastando-se a lembrança da revolução da computação desde sua invenção.

Logo, percebe-se que as técnicas de ciframento não são adequadas para longos períodos de tempo, mas são eficientes para um médio prazo, dentro da expectativa de vida dos algoritmos. Ainda há a possibilidade de uma quebra súbita, seja por uma descoberta – como a fatoração de números em primos, novas técnicas de criptoanálise ou um drástico aumento no poder computacional – a exemplo da computação quântica. Todavia, tais eventos tem baixa probabilidade de ocorrerem.

Ainda sobre algoritmos criptográficos existem duas classes de algoritmos diferenciados: algoritmos pós-quânticos e *one-time pad* (OTP). Os algoritmos pós-quânticos são projetados para sobreviver ao advento do computador quântico – que aumentaria drasticamente o poder computacional. O OTP consiste na operação de ou-exclusivo da informação com uma chave aleatória de mesmo tamanho. Tal operação torna a informação incondicionalmente segura, ou seja, mesmo com poder computacional ilimitado o atacante não consegue quebrar a cifra. Ressalta-se, conforme visto na seção 2.4.6, que o OTP pode ser visto como um caso particular do compartilhamento de segredo onde $m = n$.

Algoritmos pós-quânticos não serão abordados nesse trabalho visto que os computadores quânticos existentes ainda não possuem poder computacional suficiente para ameaçarem a criptografia atual. Além disso, existem mecanismos incondicionalmente seguros que, mesmo com tal ad-

vento, permanecerão seguros.

Já em relação ao OTP, seu emprego requer o gerenciamento de chaves criptográficas, uma vez que o sigilo da informação depende dessas chaves que, se perdidas, resultam na perda da informação sigilosa.

Shannon (1948) descreveu que para prover segurança incondicional é necessário que a chave seja, no mínimo, do mesmo tamanho do segredo e citou o OTP como exemplo. Para que o compartilhamento de segredo cumpra esse requisito as partes devem ter pelo menos o mesmo tamanho do segredo (SIMMONS, 1999). Assim, a exposição de até $m-1$ partes do segredo não revelam qualquer informação sigilosa.

Um esquema de compartilhamento de segredo é dito perfeito se os participantes (e todos os participantes não autorizados) não possuem qualquer vantagem que externos em adivinhar o segredo. O compartilhamento de Shamir (1979) é perfeito nesse sentido enquanto que o de Blakley (1979), não (SIMMONS, 1999).

Uma abordagem híbrida de ciframento e compartilhamento de segredo foi proposta por (WANG et al., 2006) na qual os documentos são cifrados por chaves simétricas e tais chaves são protegidas por chaves assimétricas (públicas) dos destinatários. O compartilhamento de segredo é utilizado para cópia de segurança (*backup*) da chave privada do destinatário. Essa proposta possui os mesmos problemas de obsolescência do ciframento.

Assim, o compartilhamento de segredo, que promove segurança incondicional desde que não se comprometa mais de $m-1$ partes, garante o sigilo da informação por longo prazo, mesmo na presença do computador quântico.

4.3 ADVERSÁRIOS X TÉCNICAS

Com as informações descritas anteriormente percebe-se que a técnica mais adequada para o sigilo por longo prazo é o compartilhamento de segredo. No trabalho de Huhnlein et al. (2009) foi mostrado como essa técnica pode ser aplicada para documentos eletrônicos. Basicamente, o documento é divido em blocos e cada bloco é tratado como um segredo do compartilhamento. Entretanto, o emprego do compartilhamento de segredo em um protocolo para distribuição das partes introduz alguns problemas.

Na seção 2.5 descreveu-se os três modelos de adversários considerados. Nesse cenário, analisa-se o compartilhamento de segredo frente a esses ataques.

Os adversários ativos atuam de forma ativa no sistema, podendo interferir a qualquer momento e de qualquer maneira. Em um protocolo de compartilhamento de segredo, poderiam alterar as partes do segredo ou mesmo interromper a distribuição para os servidores. Uma forma de defesa contra tais adversários é o emprego de mecanismos para tornar o compartilhamento de segredo verificável (Verifiable Secret Sharing).

Outro adversário no protocolo de compartilhamento de segredos é o trapaceiro, o qual pode se aproveitar dos demais servidores, iludindo-os, para então ser o único capaz de recuperar o segredo. Um modo de contornar este adversário é, também, por meio do compartilhamento de segredo verificável. Em ambos os casos de adversários, ativo e trapaceiro, a verificação das partes em relação ao segredo sem necessidade de expô-lo, denunciaria a modificação maliciosa.

Por fim, os adversários móveis são aqueles que comprometem o compartilhamento de segredo, espaçando os ataques no tempo, até obter o mínimo necessário para reconstruí-lo. Contra-medidas para tais adversários são os esquemas de renovação das partes do segredo (SHAMIR, 1979), o compartilhamento de segredo proativo (Proactive Secret Sharing) e a redistribuição do segredo.

De acordo com Shamir (1979), pode-se renovar as partes do segredo por meio da reconstrução dessas partes: mantém-se o termo independente (o segredo), geram-se novos coeficientes aleatórios e recalculam-se os pontos da função. Entretanto, percebe-se a mesma vulnerabilidade que ocorre no re-ciframento: a informação sigilosa fica exposta no sistema durante a renovação das partes do segredo.

Todavia, as demais técnicas – o compartilhamento de segredo proativo e a redistribuição de segredo, foram elaboradas para subverter essa deficiência. Ambas as técnicas podem renovar as partes do segredo compartilhado sem a reconstrução e consequente exposição do segredo. Além de diferirem no mecanismo de renovação das partes do segredo, explorando o compartilhamento de formas diferentes, essas técnicas diferem quanto à estrutura de acesso: o compartilhamento proativo requer a mesma estrutura de acesso enquanto que o protocolo de redistribuição permite mudanças nessa estrutura. Por estrutura de acesso entende-se os valores de n e m do compartilhamento de segredo, ou seja, o número de participantes total e o mínimo necessário para reconstrução do segredo. Outra desvantagem do compartilhamento proativo é que Nikov e Nikova (2005) descobriram uma vulnerabilidade no modelo de adversários móveis de Herzberg, Krawczyk e Yung (1995), a qual pode comprometer o sigilo da informação.

4.4 ANÁLISE DA ARQUITETURA DE REFERÊNCIA

Uma visão geral da arquitetura de referência foi dada na seção 3.8. Com base na análise das técnicas de confidencialidade por longo prazo e nos modelos de ataques considerados, nesta seção analisa-se de modo crítico a arquitetura de referência, proposta por Huhnlein et al. (2009).

De acordo com a visão geral, a arquitetura de referência emprega o compartilhamento de segredo de Shamir (1979) como técnica de confidencialidade. Como visto, o compartilhamento de segredo aparece como melhor mecanismo para a preservação de longo prazo, comparado as demais técnicas, como o ciframento. Entretanto, o dispositivo de *Share Re-*

newal faz com que o objeto arquivado tenha de ser remontado quando houver remoção de servidores – de modo a invalidar as partes que ele possui, e para o *Hash Tree Renewal* – se usado o *Inside Shared Mode*, caindo no mesmo problema dos mecanismos de cifragem.

Huhnlein et al. (2009) citam uma variante no modo de compartilhar as partes do segredo, gerando-as na aplicação e utilizando o resumo criptográfico do documento para construir a árvore de Merkle. Todavia, tal abordagem apenas posterga o problema, visto que em algum momento será necessário obter um novo resumo criptográfico do documento, utilizando algum algoritmo seguro, conforme descrito no procedimento de renovação da árvore de resumo criptográfico (ver seção 3.6).

Na proposta da arquitetura de referência não foi mencionado algum mecanismo de recuperação das partes no módulo *eSafe*. Com o uso do ERS pode-se saber que as partes foram alteradas ou excluídas. Todavia, não se pode recuperá-las caso um atacante altere ou exclua $(n-m+1)$ partes, fato que resultará na perda da informação sigilosa, visto que são necessárias, pelo menos, m partes para reconstruir o documento arquivado. Esse problema torna-se maior quanto menor for a diferença entre n e m (o limiar do compartilhamento).

Ainda em relação as partes do segredo, não há mecanismos de defesa contra adversários ativos e móveis. As partes podem ser comprometidas antes de chegarem aos servidores, durante a fase de (re)distribuição, e tal alteração não será detectada. Um adversário também pode obter partes de forma contínua no tempo até obter m e, assim, reconstruir o documento arquivado, desde que o faça enquanto não houver mudanças na configuração de servidores ou um *Hash Tree Renewal* se usado o *Inside Shared Mode*.

Um equívoco quanto à renovação da árvore de resumo criptográfico é mencionado na proposta. Descreve-se a renovação como sendo a geração de uma árvore de Merkle empregando novos algoritmos quando, na verdade, como visto na seção 3.6, há o processo de redução da árvore anterior e geração da nova árvore, criando uma ligação entre as informações preservadas pela árvore anterior e a nova. Tal procedimento é necessário para se manter um histórico da preservação e meios de provar, pelo processo de verificação do registro de evidência, que um determinado objeto manteve-se autêntico durante a preservação.

Na Arquitetura de Referência não são considerados arquivos (LTAs) maliciosos. Em sistemas reais é uma suposição válida que haja comprometimento e controle de forma maliciosa, ainda mais em sistemas de longo prazo, nos quais um adversário tem um tempo indeterminado para progredir. Neste trabalho, considera-se um LTA faltoso mesmo que a falta seja maliciosa (LTA malicioso).

Outra deficiência de longo prazo é o gerenciamento da identificação (IDs) dos objetos arquivados, por parte dos clientes, por meio do *Archive Token*. Essa abordagem é possível desde que exista um modo de recuperar os IDs, uma vez que clientes não possuem infraestrutura

para armazenar uma informação por longo prazo, mesmo que armazenada em *tokens* criptográficos. Uma vez perdida a identificação do documento torna-se intratável descobrir que partes armazenadas pertencem ao documento daquele cliente.

Por fim, um protocolo para distribuição e renovação das partes não foi definido na arquitetura proposta, deixando uma lacuna importante na definição do sistema. A interação entre os componentes do sistema, clientes e servidores deve ser definida e ser segura, a fim de cumprir os requisitos de um sistema de arquivamento. Alguns dos problemas descritos surgem justamente da falta de um protocolo para a arquitetura proposta.

4.5 CONCLUSÃO

Neste capítulo analisou-se as técnicas de confidencialidade e os trabalhos relacionados quanto à manutenção de suas propriedades com o tempo, a partir das definições de segurança, mostrados nos capítulos anteriores.

Igualmente, verificou-se a resistência dessas técnicas aos modelos de adversários e possíveis soluções ou alternativas. Ambas as análises basearam-se em trabalhos e discussões compilados da literatura científica.

A partir dessas análises verificou-se a arquitetura de referência descrita no capítulo anterior, avaliando-a com relação ao que ela se propõe. Encontrou-se diversos problemas e lacunas nessa arquitetura, principalmente em relação ao longo prazo, que a impedem de tornar-se uma referência em sistemas de arquivamento.

No capítulo 5 propõe-se protocolos para a preservação de longo prazo do sigilo e autenticidade de documentos eletrônicos por longo prazo com base nos estudos realizados até este capítulo.

5 PRESERVAÇÃO DO SIGILO E AUTENTICIDADE

5.1 INTRODUÇÃO

Com os conceitos e definições descritos no capítulo 2, os trabalhos relacionados estudados no capítulo 3, as análises das técnicas criptográficas a longo prazo e ataques as técnicas vistas no capítulo 4, percebeu-se uma lacuna na literatura: não há uma proposta que trate do sigilo e autenticidade de documentos eletrônicos por longo prazo.

Então, buscou-se elaborar um esquema para a preservação do sigilo e autenticidade de documentos eletrônicos a partir dos trabalhos encontrados na literatura. Assim, duas propostas foram criadas.

Um estudo inicial do protocolo G^2_{its}VSR de Gupta e Gopinath (2007) culminou no protocolo preliminar descrito na seção 5.2. Tal proposta foi apresentada no *7th European Workshop on Public Key Services, Applications and Infrastructures* (EuroPKI'10) (RAMOS et al., 2011).

Entretanto, como descrito nas seções de avaliação (seção 5.2.1) e limitações (seção 5.2.2) desse protocolo, a proposta preliminar tem limitações e custos proibitivos. Por este motivo, um novo protocolo foi proposto, que aprimora e não possui as limitações da versão preliminar e está descrito na seção 5.3.

Na seção 5.4 descreve-se os procedimentos da sintaxe do registro de evidência, complementados para utilização no protocolo proposto.

Por fim, faz-se as conclusões deste capítulo na seção 5.5.

5.2 PROTOCOLO PRELIMINAR

O principal problema da Arquitetura de Referência (HUHNLEIN et al., 2009), descrita na seção 3.8 do capítulo 3, é o mecanismo de renovação das partes, o qual expõe o segredo de forma arriscada em um servidor central de recuperação, sendo um atrativo para atacantes. Dessa forma, a vantagem de se utilizar o compartilhamento de segredo fica reduzida a segurança contra intrusões do servidor central.

De modo a aprimorar tal proposta, estudou-se os protocolos de redistribuição de segredo a fim de incorporá-los a esta arquitetura. Conforme visto na seção 3.5, Desmedt e Jajodia (1997) propuseram o protocolo para redistribuição de segredo sem reconstrução. Wong, Wang e Wing (2002), Gupta e Gopinath (2006) e, finalmente, Gupta e Gopinath (2007) aprimoraram essa proposta incorporando protocolos de segredo verificável, entre outras melhorias. Este último trabalho contém uma revisão e a respectiva correção dos protocolos anteriores e a adoção da abordagem de Pedersen (1991) de VSS, provendo segurança teórica da informação (incondicional).

A proposta preliminar, então, consiste na incorporação do protocolo G^2_{its}VSR (GUPTA; GOPINATH, 2007) à Arquitetura de Referência – um mecanismo essencial que não foi considerado nesta arquitetura, resolvendo assim os problemas de renovação das partes e adversários móveis, além de tratar da possibilidade de LTAs faltosos. Na sequência desta seção será discutido em detalhe cada um desses problemas juntamente com a solução proposta.

Na versão preliminar o protocolo G^2_{its}VSR foi adotado parcialmente. Nessa abordagem, cada parte do segredo (s_i) é redistribuída novamente por um compartilhamento de segredo. Assim, tem-se $n.n'$ partes após cada redistribuição. Ou seja, as partes temporárias, vistas na seção 3.5, são armazenadas ao invés de reconstruídas.

Conforme descrito por Gupta e Gopinath (2007), em relação as informações enviadas, o protocolo G^2_{its}VSR envia o dobro de informações comparado ao G^2_{cs}VSR, uma versão revisada do protocolo xVSR. Isso se deve à técnica de compartilhamento de segredo verificável (ver seção 3.3) empregada em cada procolo. No G^2_{cs}VSR é utilizada a técnica de Feldman (1987) enquanto que no G^2_{its}VSR (PEDERSEN, 1991) é usado. Este utiliza um valor t aleatório como segredo para um compartilhamento e cada parte desse compartilhamento é empregada como *one-time pad* (ciframento) das partes do segredo em si (k), enviando pares de partes (ou pares de subpartes) ao invés de partes (ou subpartes).

Quando todos os servidores recebem suas novas partes e toda a informação é verificada, os servidores devem apagar suas partes do período anterior. A segurança contra adversários móveis baseia-se nessa remoção e, se tal procedimento não for executado, um adversário pode obter alguma informação ou até mesmo comprometer o segredo.

> Clientes também necessitam de uma heurística para selecionar o valor do limiar m dado n servidores. Requirimos m servidores não-faltosos, e podemos tolerar no máximo $m-1$ servidores faltosos; [...] Assim, temos a restrição de que $m + (m-1) \leq n$, ou:
>
> $$m \leq \lfloor \frac{n+1}{2} \rfloor \tag{5.1}$$
>
> (WONG; WANG; WING, 2002, tradução nossa).

O emprego do protocolo G^2_{its}VSR modificado resolve o problema da renovação das partes uma vez que se pode aplicar a redistribuição quando houver mudanças estruturais, seja adicionando ou removendo um servidor da infraestrutura. Os adversários móveis podem ser mitigados aplicando a redistribuição dos segredos antes que um atacante consiga obter m partes e reconstruir o segredo S.

Já na proposta do protocolo xVSR, predecessor do G^2_{its}VSR, suporta-se LTAs maliciosos a fim de cobrir situações mais realísticas, requerendo que apenas a maioria seja confiável (*trustworthy*), ao invés de

todos os destinatários (servidores). Por fim, define-se um protocolo para o arquivo – uma lacuna existente na Arquitetura de Referência. O esquema de compartilhamento de segredo verificável adotado no protocolo (PEDERSEN, 1991) torna possível a verificação das partes pelos servidores e ainda provê segurança teórica da informação.

A manutenção da autenticidade dos documentos eletrônicos arquivados é realizada por meio do ERS (ver seção 3.6). Os documentos são divididos em partes pelo compartilhamento de segredo e essas partes são usadas na geração da árvore de Merkle (1980). Ao concluir a construção da árvore o resumo criptográfico da raiz recebe um carimbo do tempo, atestando a data em que a árvore foi produzida. Conforme o ERS, a árvore é reduzida e, então, são gerados os registros de evidência para cada parte do segredo (do documento).

Para exemplificar, seja um cliente C compartilhando um documento S com limiar $(m = 2, n = 4)$, gerando as partes (s_1, s_2, s_3, s_4). Cada servidor i componente do protocolo recebe a parte s_i. Uma árvore de Merkle é criada a partir do resumo criptográfico dessas partes (h_1, h_2, h_3, h_4), considerando a ordem binária $h_1 < h_2 < h_3 < h_4$ e $h_{12} < h_{34}$. A figura 5.1 ilustra essa árvore.

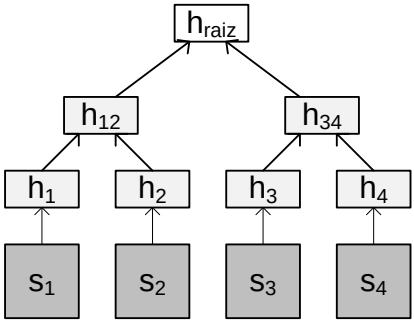

Figura 5.1: Árvore de Merkle construída do resumo criptográfico das partes s_1, s_2, s_3 e s_4.

De acordo com o ERS (ver seção 3.6), os registros de evidência (ER) são gerados a partir da redução da árvore de Merkle. Para as partes s_1 e s_2 a árvore reduzida compreende os nós h_1, h_2 e h_{34}. Já para as partes s_3 e s_4 tem-se os nós h_3, h_4 e h_{12}. Os nós que possuem o mesmo nó pai (por exemplo, h_1 e h_2 tem o nó h_{12} como pai), também possuem a mesma árvore reduzida. A figura 5.2 ilustra tais árvores.

Percebe-se que a árvore reduzida contém o mínimo de nós necessários para reconstrução da árvore de Merkle. No primeiro caso, a concatenação (simbolizado por $|$) e obtenção do resumo criptográfico (H) dos nós h_1 e h_2 ($H(h_1|h_2)$) resulta no resumo h_{12} que, concatenado com o nó h_{34} e obtido o resumo do valor resultante, obtém-se o resumo h_{raiz}. Do mesmo modo, $H(h_3|h_4) = h_{34}$ e $H(h_{12}|h_{34}) = h_{raiz}$. Com o resumo

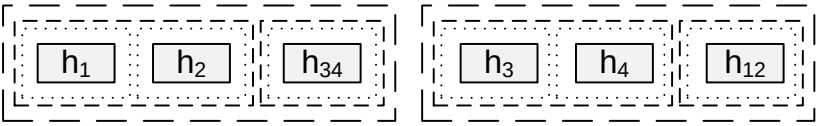

Figura 5.2: Árvores reduzidas para as partes s_1 e s_2 (esq.) e para as partes s_3 e s_4 (dir.).

do nó raiz pode-se verificar a validade do carimbo do tempo presente no ER e verificar se o resumo contido no carimbo é o mesmo resumo calculado e, em caso afirmativo, comprova-se a autenticidade desses nós. Igualmente, comparando o resumo criptográfico dos nós contidos no ER com o resumo calculado das partes (s_i), verifica-se que os nós são, de fato, os mesmos preservados pelo registro de evidência. Reconstruindo o documento a partir dessas partes comprova-se a preservação e autenticidade do documento.

Agora, considere que as partes compartilhadas (s_1, s_2, s_3, s_4), referentes ao documento S, sejam redistribuídas com limiar ($m' = 2, n' = 3$), supondo que um dos servidores ($i = 4$) foi desligado para manutenção. Como ainda restam três servidores e três deles são suficientes para reconstruir o segredo ($m = 2 \leq 3$), a falta de um servidor não compromete a execução do protocolo.

Cada servidor i aplica um compartilhamento de segredo em s_i, originando as subpartes s_{i1}, s_{i2}, s_{i3}, resultando no subconjunto (s_{11}, s_{12}, s_{13}),(s_{21}, s_{22}, s_{23}),(s_{31}, s_{32}, s_{33}). Ou seja, cada parte primária (s_1, s_2, s_3) é tratada como um segredo e compartilhada com o limiar escolhido. As n partes iniciais, excluindo-se a parte s_4 pois o servidor estava desligado, deram origem a novas $(n-1) \cdot n' = 9$ subpartes. Percebe-se que houve uma mudança na estrutura de acesso e que o protocolo pode realizá-la sem reconstruir o segredo.

Cada servidor i mantém consigo a subparte s_{ii}, enviando as restantes para os demais servidores. Ao final desta etapa cada servidor i terá consigo as subpartes s_{1i}, s_{2i}, s_{3i}. Após a execução correta do protocolo as partes primárias (do período anterior) são removidas do sistema, evitando assim os adversários móveis. A figura 5.3 ilustra a redistribuição.

Para comprovar a autenticidade das subpartes é necessário que os servidores executem uma reconstrução do compartilhamento de segredo, de modo a retornar as partes primárias. Para isso, devem combinar pelo menos duas (m') subpartes de cada índice, por exemplo, pelo menos duas subpartes entre s_{11}, s_{12} e s_{13}) para o índice 1. Reconstruindo essas subpartes recupera-se as partes primárias que, por sua vez, podem ser comprovadas como descrito anteriormente. A figura 5.4 ilustra a reconstrução.

Ainda, considere que dois novos servidores foram adicionados à infraestrutura e uma nova redistribuição ocorreu, com limiar $m'' = 3$, $n'' = 5$, a partir a redistribuição anterior. As sub-subpartes resultantes dessa execução seriam $(s_{111}, s_{112}, s_{113}, s_{114}, s_{115}), (s_{121}, s_{122}, \ldots,$

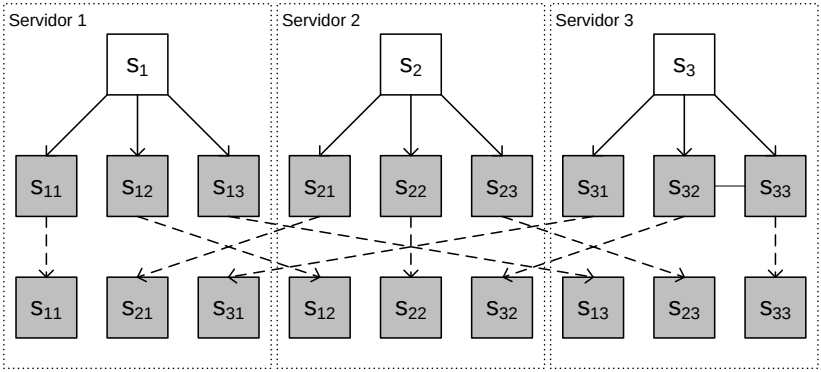

Figura 5.3: Redistribuição das partes s_1, s_2 e s_3.

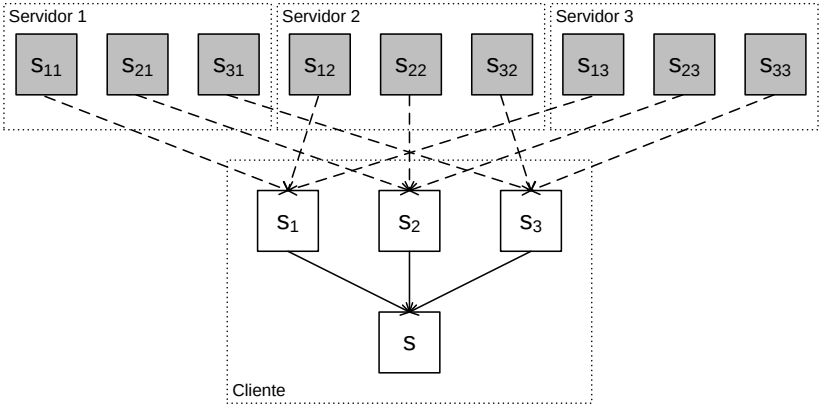

Figura 5.4: Reconstrução das partes s_1, s_2 e s_3 e reconstrução do documento S a partir das partes reconstruídas.

s_{125}), ($s_{131}, s_{132}, \ldots, s_{135}$), ($s_{211}, s_{212}, \ldots, s_{215}$), $\ldots$, ($s_{331}, s_{332}, s_{333}, s_{334}, s_{335}$). Do mesmo modo que no período anterior, cada servidor i mantém consigo a subparte s_{iii} e envia as restantes para os demais servidores. Ao final desta etapa, cada servidor i terá consigo as subpartes s_{11i}, $s_{12i}, s_{13i}, s_{21i}, s_{22i}, s_{23i}, s_{31i}, s_{32i}, s_{33i}$. Após a correta execução, foram geradas $(n-1) \cdot n' \cdot n'' = 45$ novas sub-subpartes e $(n-1) \cdot n' = 9$ subpartes, do período anterior, podem ser removidas do sistema.

Para comprovar a autenticidade das sub-subpartes é necessário que os servidores executem duas etapas de reconstrução do compartilhamento de segredo, de modo a retornar, primeiramente as subpartes, e então as partes primárias. Para tanto, devem combinar pelo menos três (m'') sub-subpartes de cada índice, por exemplo, pelo menos três sub-subpartes entre $s_{111}, s_{112}, s_{113}, s_{114}$ e s_{115}) para o índice 1. Reconstruindo essas sub-subpartes recupera-se as subpartes. Executando a recuperação das

subpartes descrita anteriormente, reconstrói-se as partes primárias que, finalmente, podem ser comprovadas como já descrito.

Em qualquer momento o cliente pode reconstruir o documento S realizando as sucessivas reconstruções, seja após a etapa inicial ou uma redistribuição, conforme descrito acima, e verificar a autenticidade desse documento por meio da verificação das partes primárias, da reconstrução da árvore de Merkle e da verificação do carimbo do tempo aposto na raiz.

Os valores de $m = 2$, $m' = 2$ e $m'' = 3$ foram escolhidos respeitando-se a restrição descrita na equação 5.1 em relação aos valores de $n = 4$, $n' = 3$ e $n'' = 5$. Por meio dessa equação, verifica-se que $2 \leq \lfloor \frac{4+1}{2} \rfloor$, $2 \leq \lfloor \frac{3+1}{2} \rfloor$ e $3 \leq \lfloor \frac{5+1}{2} \rfloor$.

Apresentado o protocolo e exemplificado de forma genérica seu funcionamento, parte-se para a avaliação desse protocolo, na seção seguinte, definindo métricas e utilizando valores concretos. Obtém-se, assim, dados para avaliar o protocolo proposto.

5.2.1 Avaliação

A partir das definições do protocolo preliminar descreve-se duas equações para avaliar os custos de armazenamento após as etapas inicial e de redistribuição do protocolo. Por custos de armazenamento entende-se o espaço em disco necessário para armazenar as partes do segredo, calculadas durante a execução do protocolo. Nestas equações os custos são globais, ou seja, referentes a todas as partes do compartilhamento. Para se obter um custo por servidor pode-se dividir pelo número de servidores utilizados. Ainda, são considerados somente os custos e operações referentes ao compartilhamento do segredo, excluindo-se os custos referentes ao mecanismo de compartilhamento verificável.

Quando um cliente envia um documento F aos servidores este é quebrado em blocos de t bytes e uma operação de compartilhamento de segredo (m, n) é aplicada para cada bloco. O tamanho dos blocos é definido por $t = \lceil \frac{|F|}{|l|} \rceil$, sendo $|F|$ o tamanho do arquivo e $|l|$ o tamanho do módulo, uma vez que o compartilhamento de segredo requerer que as partes tenham pelo menos o mesmo tamanho da informação para atingir a segurança incondicional. Apos isso é aplicada uma operação de compartilhamento de segredo para cada um dos b blocos (b operações) e o custo de armazenamento c será:

$$c = b \cdot n \cdot t \tag{5.2}$$

Em uma redistribuição já se tem os blocos fragmentados na etapa inicial. Assim, uma operação de compartilhamento de segredo, com limiar (m', n'), será executada para cada b blocos de n partes ($n \cdot b$ operações) e o novo custo de armazenamento (c') será $c' = n'(b \cdot n \cdot t) - c$, pois com a remoção das partes do período anterior, após a correta execução do protocolo de redistribuição, tem-se a liberação de c bytes de espaço em

disco. Como $b \cdot n \cdot t = c$, tem-se que $c' = n' \cdot -c$ ou:

$$c' = c(n' - 1) \tag{5.3}$$

Ou seja, cada redistribuição aumenta os custos em $n' - 1$ vezes o custo de armazenamento do período anterior (c).

Nas equações 5.2 e 5.3 foram utilizadas as variáveis m' e n' uma vez que a estrutura de acesso pode ser alterada em cada redistribuição, desde que $n' \geq m$ para permitir a reconstrução das partes do período anterior. Nos casos em que o tamanho do documento não for múltiplo do tamanho do módulo, o valor obtido por estas fórmulas será aproximado, resultando em um custo maior do que o real. Entretanto, a diferença dos custos aproximado (c_a) e real (c_r) será, no máximo, $n - 1$ (ou $n' - 1$) vezes o tamanho do módulo ($|l|$), ou $c_a - c_r \leq (n - 1) \cdot |l|$.

Definidas as equações para avaliação dos custos de armazenamento elabora-se um exemplo para avaliação do protocolo preliminar. Seja o compartilhamento de um arquivo de 9.888 bytes, um módulo primo de $t = 48$ bytes e $n = 3$ servidores. Então, tem-se $b = 206$ blocos e o custo resultante, de acordo com a equação 5.2, será de $c = 29.664$ bytes após a distribuição (etapa inicial). Nota-se que nesse e nos demais exemplos o tamanho do arquivo é múltiplo do tamanho do módulo e, portanto, os custos calculados são iguais aos custos reais.

Considerando que seja mantida a mesma estrutura de acesso, para a primeira redistribuição teria-se o custo de $c' = 29.664 \cdot (3 - 1) = 59.328$ bytes ($57,94$ KiB[1]), de acordo com a equação 5.3. A tabela 5.1 e a figura 5.5 ilustram os custos de armazenamento após cinco redistribuições (rodadas).

Tabela 5.1: Custos de armazenamento (c') após cinco redistribuição.

rodadas	1	2	3	4	5
c' (em KiB)	$57,94$	$202,78$	$579,38$	$1.767,09$	$5.272,32$

Outra avaliação é se o tamanho do módulo tem impacto significativo no custo de armazenamento. A tabela 5.2 ilustra os custos de armazenamento (c') quando se escolhe módulos de diferentes tamanhos, considerando que sejam mantidas as demais configurações. O tamanho do módulo é descrito em bytes e os custos de armazenamento em kibibytes (KiB).

Pode-se inferir da tabela 5.2 que os custos de armazenamento crescem rapidamente e que o tamanho do módulo não tem grande impacto nesses custos, conforme esperado. Entretanto, reduzindo-se o tamanho do módulo ocorre um aumento no número de blocos e, consequentemente,

[1] 1 kibibyte (KiB) = 1024 bytes, de acordo com o padrão IEC 60027-2 do *International Electrotechnical Commission* (IEC). Texto explicativo disponível em <`http://www.iec.ch/zone/si/si\use_bytes.htm`>

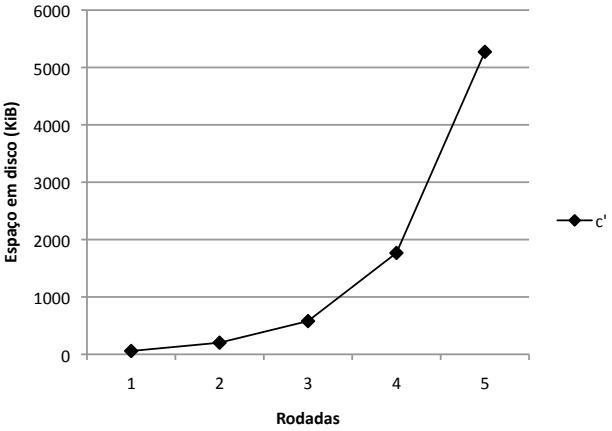

Figura 5.5: Custos de armazenamento (c') após cinco redistribuição.

Tabela 5.2: Custos de armazenamento (c') após cinco rodadas de redistribuição com diferentes tamanhos de módulo.

módulo	c'_1	c'_2	c'_3	c'_4	c'_5
6	$57,9$	$202,66$	$579,02$	$1.766,02$	$5.269,11$
12	$57,94$	$202,78$	$579,38$	$1.767,09$	$5.272,32$
24	$57,94$	$202,78$	$579,38$	$1.767,09$	$5.272,32$
48	$57,94$	$202,78$	$579,38$	$1.767,09$	$5.272,32$
50	$58,01$	$203,03$	$580,08$	$1.769,24$	$5.278,71$

um aumento no número de operações de compartilhamento de segredo que serão executadas ($n' \cdot b$ operações). A tabela 5.3 e a figura 5.6 mostram o número de operações de compartilhamento de segredo por redistribuição quando diferentes tamanhos de módulos (em bytes) são utilizados.

Tabela 5.3: Operações de compartilhamento de segredo por rodada (c'_i) com módulos de diferentes tamanhos (em bytes).

módulo	c'_1	c'_2	c'_3	c'_4	c'_5
6	4.944	14.832	44.496	133.488	400.464
12	2.472	7.416	22.248	66.744	200.232
24	1.236	3.708	11.124	33.372	100.116
48	618	1.854	5.562	16.686	50.058
50	594	1.782	5.346	16.038	48.114

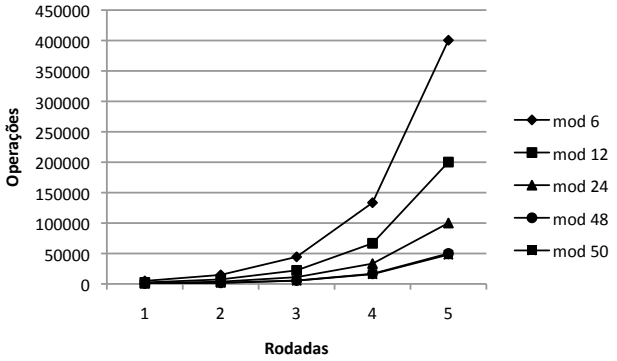

Figura 5.6: Operações de compartilhamento de segredo por rodada com módulos de diferentes tamanhos.

5.2.2 Limitações

A partir das avaliações da seção anterior percebe-se que o crescimento do número de partes no sistema é proporcional ao novo número dos servidores em cada redistribuição (n'). Como a tendência é que se tenha, cada vez mais, novos documentos incorporados ao sistema, pode-se saturar rapidamente a infraestrutura de armazenamento devido ao alto custo do protocolo.

Foi mostrado que a redistribuição não afeta a ligação das partes com a árvore de Merkle e, consequentemente, a autenticidade do documento compartilhado. Contudo, conforme visto na descrição do protocolo preliminar (ver seção 5.2) é necessário reconstruir t vezes, sendo t o número de redistribuições executadas, as subpartes do sistema até se obter as partes primárias para, assim, verificar a pertinência na árvore de Merkle. Todavia, este processo é realizado pelo cliente que, de qualquer modo, precisa reconstruir essas partes para obter o documento, não constituindo um problema mais grave, exceto pela quantidade de operações necessárias para reconstrução.

Entretanto, a maior limitação desta proposta é quando for necessário atualizar a árvore de resumo criptográfico empregado na árvore de Merkle (*Hash Tree Renewal*). Neste caso, segundo o ERS (ver seção 3.6), deve-se obter o resumo criptográfico das partes existentes no sistema com um novo algoritmo de resumo criptográfico. Mas, as partes primárias foram removidas, a fim de evitar os adversários móveis, e não mais existem no sistema, e sim subpartes dessas partes. Para renovar a árvore de resumo criptográfico é preciso reconstruir t vezes as subpartes existentes, sendo t o número de redistribuições executadas, do mesmo modo que para provar a autenticidade do documento. Porém, a diferença nesse caso é que os servidores serão os executores da tarefa, o que irá requerer conhecimento

de que partes compõem o segredo e um ambiente seguro para realizar esta atividade. O ambiente seguro é necessário pois um adversário móvel poderia se aproveitar desse momento para conseguir as partes que não possui, eventualmente obtendo o mínimo necessário para reconstruir o segredo.

Por exemplo, suponha que um adversário tenha consigo duas de quatro partes primárias, que o limiar utilizado foi de $(m = 3, n = 4)$, e que ocorreram t redistribuições, sendo $t > 0$. No processo de renovação da árvore de resumo criptográfico será necessário retornar para as partes primárias e, se o ambiente de reconstrução não for seguro o suficiente, o adversário poderá obter uma nova parte primária e, assim, reconstruir o documento sigiloso.

Tal procedimento é equivalente a reconstrução do segredo em um servidor para posterior renovação criptográfica e, portanto, não melhora os problemas de longo prazo encontrados na literatura, por exemplo, em sistemas que utilizam ciframento para garantir o sigilo de documentos eletrônicos, conforme visto no capítulo 4.

As t reconstruções são utilizadas apenas para obter novamente as partes primárias e calcular o resumo criptográfico dessas partes, utilizando um algoritmo seguro. As redistribuições subsequentes continuam do ponto em que pararam, ou seja, utilizando as partes do período t.

Um aprimoramento desse protocolo de modo a diminuir o número de reconstruções necessárias para se obter novamente as partes primárias é a introdução de um ciclo, utilizando apenas uma redistribuição. Nesse caso, antes de cada redistribuição há uma reconstrução das partes primárias em um ambiente seguro e posterior redistribuição. Como os coeficientes são aleatórios as partes geradas em cada ciclo de redistribuição serão sempre diferentes, mas reconstruindo as mesmas partes primárias. Com tal adaptação, para comprovar a autenticidade das partes primárias, e posteriormente do documento, e para a renovação da árvore de resumo criptográfico, seria necessária apenas uma reconstrução.

A introdução de ciclos na redistribuição também diminui os custos de armazenamento visto que, no máximo, haverá $n \cdot n'$ subpartes armazenadas. Todavia, o emprego de ciclos torna mais frequente a reconstrução das partes primárias e, consequentemente, mais atrativa aos adversários.

5.3 NOVO PROTOCOLO

Dispondo das avaliações e limitações da proposta anterior e com o uso do protocolo G^2_{its}VSR (GUPTA; GOPINATH, 2007) completo, buscou-se uma nova abordagem que evoluísse a proposta preliminar e, principalmente, resolvesse as limitações apostas.

No protocolo G^2_{its}VSR as $n \cdot n'$ subpartes criadas durante a redistribuição são reordenadas após a verificação de sua corretude, e então reconstruídas, de modo a permanecer somente n' partes, sendo estas partes completamente disjuntas das partes iniciais. Estas são removidas ao

final da execução do processo de redistribuição. Contudo, as n' partes reconstroem o mesmo segredo, conforme proposto por Desmedt e Jajodia (1997).

Assim, tem-se um número fixo de partes a cada redistribuição (n') que reconstroem o documento (o segredo). Todavia, como as partes são alteradas e não se consegue voltar para as partes iniciais, perde-se a autenticidade obtida por meio da árvore de Merkle, visto que os resumos criptográficos preservados são das partes iniciais, que não mais existem no sistema.

Esse problema é similar ao existente na renovação da árvore de resumo criptográfico do protocolo preliminar, apresentado na seção 5.2. Entretanto, naquele protocolo era possível reconstruir as partes primárias e verificá-las junto ao registro de evidência. Neste protocolo esse procedimento não é possível.

Este cenário motivou a utilização parcial do protocolo G^2_{its}VSR no protocolo preliminar vislumbrando que as modificações propostas resolveriam o problema, apesar dos altos custos. Como descrito anteriormente, de fato, tal problema foi contornado, mesmo não sendo uma solução ótima.

De modo a resolver esse problema, ou seja, permitir a verificação da autenticidade das partes primárias e aprimorar o protocolo preliminar, propõe-se a utilização da técnica de compartilhamento de segredo verificável utilizada no protocolo G^2_{its}VSR (técnica de Pedersen (1991)), apresentada na seção 3.3. Com essa proposta, ao invés de se utilizar o resumo criptográfico das partes primárias do segredo, utiliza-se o resumo da testemunha do segredo ($g^k h^t$).

O novo protocolo será detalhado a seguir, partindo do protocolo G^2_{its}VSR, e adaptando-o de modo a atender o requisito de autenticidade dos documentos sigilosos. Os detalhes sobre a autenticidade serão tratados na seção 5.4.

O protocolo G^2_{its}VSR foi complementado para:

i. incluir um módulo de adaptação da entrada, dividindo o arquivo do cliente em blocos de tamanho fixo;

ii. incluir a geração da árvore de Merkle de forma a contemplar o protocolo ERS e, assim, garantir o atendimento do requisito de autenticidade em longo prazo do documento sigiloso;

iii. explicitamente remover as partes anteriores que não são mais necessárias após cada redistribuição; e

iv. que o cliente também tenha acesso à estrutura ERS na etapa de reconstrução do segredo.

Serão descritos nas seções seguintes os três procedimentos que compõem o protocolo: inicial, redistribuição e recuperação. O protocolo

G^2_{its}VSR e suas fases já foram descritos na seção 3.5. Essa nova descrição deve-se as adaptações enumeradas acima.

As preliminares do novo protocolo são idênticas as preliminares do protocolo G^2_{its}VSR, descritas na seção 3.5.

Em cada um dos procedimentos estão definidos os seus passos e, ao final da listagem, explica-se as execuções realizadas.

5.3.1 Início

Este procedimento consiste no compartilhamento de um documento F para n servidores, por um cliente C, sendo m servidores necessários para reconstrução desse documento ($m \leq n$), formando uma estrutura de acesso (m, n). Essa distribuição ocorre da seguinte forma:

1. C divide F em blocos de tamanho $|p|$ formando um conjunto K contendo $\lceil \frac{|F|}{|p|} \rceil$ elementos;

2. Para todo $k \in K$, C escolhe um número t aleatório de $\mathbb{Z}_p$ e dois polinômios $a(i) = k + a_1 i + \cdots + a_{m-1} i^{m-1}$ e $b(i) = t + b_1 i + \cdots + b_{m-1} i^{m-1}$ para calcular as partes $s_i = a(i)$ e $t_i = b(i)$, de k e t, agrupa os valores $y_i = a(i)$ em i conjuntos S, agrupa os valores $y_i = b(i)$ em i conjuntos T, e os envia (S_i, T_i) para o servidor i;

3. Para todo $k \in K$, C usa os geradores g e h para calcular $g^k, g^{a_1}, \cdots, g^{a_{m-1}}$ e $h^t, h^{b_1}, \cdots, h^{b_{m-1}}$, calcular as testemunhas $g^k h^t$, $g^{a_1} h^{b_1}$, $\cdots$, $g^{a_{m-1}} h^{b_{m-1}}$ e enviá-las para todos os servidores via *broadcast*;

4. Cada servidor verifica, $\forall s \in S, t \in T$, se:

$$g^{s_i} h^{t_i} \equiv g^k h^t \prod_{l=1}^{m-1} (g^{a_l} h^{b_l})^{i^l} \tag{5.4}$$

   - Se a verificação proceder o servidor i mantém as partes (s_i, t_i);
   - Caso a verificação falhe o servidor i registra uma reclamação do cliente e inicia-se o sub-protocolo de reclamação. Esse sub-protocolo tem por objetivo verificar quem está sendo desonesto, se é o cliente ou o servidor:

(a) o cliente envia pelo canal de *broadcast* as partes $(\hat{s}_i, \hat{t}_i)$ enviadas ao servidor i;

(b) os servidores verificam as partes reveladas pela equação 5.4

   - Se a verificação proceder os demais servidores marcam o servidor i como desonesto;

- Caso a verificação falhe o cliente é marcado como desonesto e o protocolo é abortado.

5. Para todo $k \in K$, cada servidor obtém um resumo criptográfico das testemunhas $g^k h^t$ e gera um ERS (ver seção 5.4);

6. A maioria honesta dos servidores recebeu o par (S_i, T_i) correto, possui as testemunhas $g^k h^t$ e um ERS gerado a partir de tais testemunhas. Esses dados são armazenados por cada servidor.

O documento F é fracionando em blocos, considerados como um segredo k do compartilhamento de segredo, obtendo-se n partes para cada bloco. Tais partes são, então, agrupadas de modo a formar n pares (n partes do segredo). O segredo t compartilhado no protocolo é um número aleatório usado como *one-time pad* para o segredo k. Também, uma testemunha ($g^k h^t$) é calculada para cada bloco e a árvore de Merkle a ser reduzida e transformada em um ERS é criada a partir do resumo criptográfico dessas testemunhas.

Conforme pontuou Gupta e Gopinath (2007), no sub-protocolo de reclamação há a "revelação" de uma parte do segredo para verificação de honestidade das entidades envolvidas (cliente e servidores). Entretanto, esse protocolo só é executado em caso de falha na verificação das partes e, consequentemente, uma das entidades é desonesta e já conhecerá a parte revelada. Portanto, não há efetivamente uma revelação no sub-protocolo. Contanto que não haja m desonestos, o procedimento é executado corretamente e não há quebra de sigilo.

Todavia, deve-se registrar o número de desonestos na execução do protocolo para ser utilizado na política de redistribuição, pois se houver $m-1$ desonestos, $m-1$ partes serão conhecidas e com apenas mais uma parte o segredo poderá ser reconstruído.

Ainda sobre o sub-protocolo de reclamação, segundo Gupta e Gopinath (2007), denotou-se as partes (s_i, t_i) como $(\hat{s}_i, \hat{t}_i)$ pois elas podem ser diferentes das partes do passo anterior, visto que o cliente pode tê-las modificado caso seja desonesto.

A partir da fase inicial o sistema se encarrega da manutenção da confidencialidade e autenticidade do arquivo F sem intervenção do cliente. A qualquer momento o cliente pode recuperar seu arquivo no sistema por meio da fase de recuperação.

5.3.2 Redistribuição

Neste procedimento cada servidor i (emissor) redistribui o par (S_i, T_i) para n' servidores (receptores), ou seja, da estrutura de acesso (m, n) para (m', n'). Os passos dessa etapa de redistribuição são:

1. Para todo $s \in S$ e $t \in T$, cada emissor i escolhe dois polinômios $a'_i(j) = s_i + a'_{i1} j + \cdots + a'_{i(m'-1)} j^{m'-1}$ e $b'_i(j) = t_i + b'_{i1} j + \cdots +$

$b'_{i(m'-1)} j^{m'-1}$ para calcular as partes $\hat{s}_{ij} = a'_i(j)$ e $\hat{t}_{ij} = b'_i(j)$, de s_i e t_i, agrupa os valores $y'_i = a'_i(j)$ em i conjuntos $\hat{S}$, agrupa os valores $y'_i = b'_i(j)$ em i conjuntos $\hat{T}$, e os envia $(\hat{S}_i, \hat{T}_i)$ para o servidor j;

2. Para todo $s \in S$, cada emissor i usa os geradores g e h para calcular $g^{s_i}, g^{a'_1}, \cdots, g^{a'_i(m'-1)}$ e $h^{t_i}, h^{b'_i 1}, \cdots, h^{b'_i(m'-1)}$, calcular as testemunhas $g^{s_i} h^{t_i}$, $g^{a'_{i1}} h^{b'_{i1}}$, $\cdots$, $g^{a'_{i(m'-1)}} h^{b'_{i(m'-1)}}$ e enviá-las para todos os receptores via *broadcast*;

3. Cada receptor j verifica, $\forall s \in \hat{S}, t \in \hat{T}$, se:

$$\forall i, g^{\hat{s}_{ij}} h^{\hat{t}_{ij}} \equiv g^{s_i} h^{t_i} \prod_{l=1}^{m'-1} (g^{a'_{il}} h^{b'_{il}})^{j^l} \tag{5.5}$$

   - Se a verificação proceder o servidor j mantém as subpartes $(\hat{s}_{ij}, \hat{t}_{ij})$;
   - Caso a verificação falhe o servidor j registra uma reclamação do servidor i e inicia-se o sub-protocolo de reclamação:

   (a) o servidor i envia pelo canal de *broadcast* as subpartes $(\hat{\hat{s}}_{ij}, \hat{\hat{t}}_{ij})$ enviadas ao servidor j;

   (b) os servidores verificam as partes reveladas pela equação 5.5
      - Se a verificação proceder os demais servidores marcam o servidor j como desonesto;
      - Caso a verificação falhe o servidor i é marcado como desonesto.

4. Cada receptor utiliza o algoritmo $\mathcal{A}$ para formar uma sequência $\mathcal{B}_u$ com m-subconjuntos de U servidores (número de emissores não marcados), onde $u = 1, \cdots, M$, $1 \leq M \leq N$ e N sendo "U escolhe m";

5. Cada emissor envia por *broadcast* as testemunhas $g^k h^t$. Pela maioria honesta, os receptores terão as testemunha corretas para realizar a verificação no passo seguinte;

6. Os receptores verificam em sucessivos $\mathcal{B}_u$, $\forall s \in S, t \in T$, a validade das partes por:

$$g^k h^t \equiv \prod_i (g^{s_i} h^{t_i})^{b_i} \text{ onde } b_i = \prod_{l \in \mathcal{B}_u, l \neq i} \frac{l}{l-i} \tag{5.6}$$

7. Cada receptor j calcula seu novo par de partes, $\forall \hat{s} \in \hat{S}, \hat{t} \in \hat{T}$, usando as subpartes do conjunto de emissores B_u, pela fórmula:

$$s'_j = \sum_{i \in \mathcal{B}_u} b_i \hat{s}_{ij} \text{ e } t'_j = \sum_{i \in B} b_i \hat{t}_{ij} \text{ onde } b_i = \prod_{l \in \mathcal{B}_u, l \neq i} \frac{l}{l-i}$$

   e armazena o par (S'_j, T'_j) e as testemunhas $g^k h^t$;

8. Cada receptor j remove o par (S_j, T_j) da rodada anterior e o par temporário $(\hat{S}_j, \hat{T}_j)$.

Os n pares armazenados são redistribuídos um a um formando $n \cdot n'$ pares temporários. O valor t correspondente do par é utilizado como *one-time pad* ao invés de ser gerado aleatoriamente como no procedimento inicial. Cada par temporário é verificado se pertence ao par inicial, por meio das testemunhas, e se está integro em relação aos valores das testemunhas armazenados. Após as rodadas de verificação procede-se a construção dos novos pares (S', T') por meio da reconstrução dos pares temporários. Ao fim, tem-se novos pares (n' pares), diferentes dos pares iniciais, mas que reconstroem o mesmo segredo na fase de reconstrução.

O sub-protocolo de reclamação é regulado pela maioria dos servidores que são considerados honestos. Nenhum servidor é removido do sistema, mas os marcados como desonestos terão suas partes ignoradas durante o procedimento. Nota-se que o protocolo executa corretamente desde que se tenha, no máximo, $m - 1$ servidores desonestos.

O conjunto $\mathcal{B}_u$ é formado pelos servidores que não foram marcados como desonestos e cujo par satisfaz a equação 5.6.

Ao final do procedimento cada receptor remove os pares (S_j, T_j) e temporário $(\hat{S}_j, \hat{T}_j)$ gerados, respectivamente, no procedimento anterior (inicial ou redistribuição) e durante a redistribuição (pares temporários gerados antes dos passos de reordenação e reconstrução). Tal etapa, constante no protocolo de Desmedt e Jajodia (1997) não é descrita nos protocolos de Wong, Wang e Wing (2002), Gupta e Gopinath (2006) e Gupta e Gopinath (2007).

5.3.3 Reconstrução

Seja C um cliente cujo documento F foi (re)distribuído e que deseja obtê-lo novamente. Os passos para reconstrução desse documento, a partir da estrutura de acesso (m, n), são:

1. C requisita o documento F. Cada servidor i envia seu par (S_i, T_i) para C;

2. C obtém as testemunhas $g^k h^t$ pela maioria dos servidores;

3. C utiliza o algoritmo $\mathcal{A}$ para formar uma sequência $\mathcal{B}_u$ com m-subconjuntos de n servidores, onde $u = 1, \cdots, M$, $M \leq N$ e N sendo "n escolhe m";

4. C verifica em sucessivos $\mathcal{B}_u$, $\forall s \in S, t \in T$, a validade das partes por:

$$g^k h^t \equiv \prod_i (g^{s_i} h^{t_i})^{b_i} \text{ onde } b_i = \prod_{l \in \mathcal{B}, l \neq i} \frac{l}{l-i}$$

5. C reconstrói o documento F, usando o primeiro $\mathcal{B}_u$ para o qual o teste aprovar, pela equação:

$$F = \bigcup_j \sum_i s_{ij} b_i \text{ onde } b_i = \prod_{l \in B_u, l \neq i} \frac{l}{l-i}$$

6. C obtém o ERS dos servidores e verifica os sucessivos ERS até encontrar um que esteja válido (ver seção 5.4).

Cada par é recuperado do sistema e verificado por meio das testemunhas $g^k h^t$ obtidas pela maioria dos servidores. Com a verificação das testemunhas o cliente tem a prova de que os pares recebidos fazem parte do documento a ser reconstruído. A seguir, verifica-se o ERS por meio do resumo criptográfico das testemunhas. Assim, o cliente tem a prova de que os pares recebidos e, consequentemente, o documento a ser reconstruído são autênticos. Por fim, o cliente obtém o documento por meio da concatenação (representada pela operação de união) dos blocos reconstruídos no compartilhamento de segredo.

Nota-se que os valores do conjunto T não são reconstruídos uma vez que tais dados são empregados apenas como *one-time pad* para as partes do segredo.

5.4 REGISTRO DE EVIDÊNCIA

A sintaxe do registro de evidência (ERS), vista na seção 3.6 e proposta na RFC 4998 (BRANDNER; PORDESCH; GONDROM, 2007), é um mecanismo de preservação da autenticidade por longo prazo por meio de renovações realizadas ao longo do tempo. Conforme descrito, a proposta baseia-se no uso das testemunhas obtidas na execução do protocolo como representação das partes que compõem o documento.

Os procedimentos do ERS serão descritos a seguir, sendo complementados para utilizar o resumo criptográfico das testemunhas ao invés do resumo do documento. Tais procedimentos são: geração do registro de evidência; renovação do carimbo do tempo; e renovação da árvore de resumo criptográfico e verificação do registro de evidência.

5.4.1 Geração do Registro de Evidência

Como mostrado na etapa *Início* do protocolo proposto (ver seção 5.3.1), o registro de evidência é gerado ao final desse passo por cada um dos servidores do sistema.

As etapas para a geração do registro de evidência são basicamente as mesmas definidas na RFC, exceto que os objetos serão as testemunhas $g^k h^t$ de cada bloco do documento a ser mantido. Então, tem-se os seguintes passos:

1. obter as testemunhas das partes para serem carimbadas;

2. escolher um algoritmo de resumo criptográfico seguro e obter os resumos das testemunhas (serão as folhas da árvore);

3. caso tenha mais de um resumo criptográfico, agrupá-los e ordenálos em ordem binária ascendente. Repetir até restar um único resumo;

4. obter um carimbo do tempo para esse resumo da raiz.

A partir da obtenção desses dados pode-se gerar o registro de evidência para fins de armazenamento, com base na estrutura ASN.1. A geração do registro de evidência pode ser realizada por diferentes políticas, por exemplo, quando houver um certo número de resumos criptográficos disponíveis. Entretanto, o sistema deve armazenar de forma segura tais dados enquanto não gerar o registro de evidência.

A geração de um registro de evidência por cada servidor justifica-se dada a imprevisibilidade de quando é necessário realizar as renovações. Assim, escolhendo-se diferentes algoritmos, diferentes autoridades de carimbo do tempo e até mesmo diferentes autoridades certificadoras raízes para cada um dos servidores, tem-se uma menor probabilidade de perda da autenticidade e evita-se um ponto único de falha.

5.4.2 Renovação do Carimbo do Tempo

A renovação do carimbo do tempo (*Time-Stamp Renewal*) é o procedimento pra renovação do carimbo do tempo que contém o resumo criptográfico da raiz da árvore de Merkle. Este carimbo deve ser renovado antes que ocorra uma expiração, revogação ou comprometimento de um dos algoritmos empregados em qualquer componente do caminho de certificação da autoridade de carimbo do tempo e da carimbadora.

Em relação ao procedimento descrito no ERS não é necessário fazer qualquer alteração no protocolo proposto. O conteúdo do campo *timeStamp* da estrutura *ArchiveTimeStamp* deve ter seu resumo criptográfico calculado e ser carimbado por um novo carimbo do tempo, gerando uma nova estrutura *ArchiveTimeStamp* que deve ser adicionada de

forma cronológica ascendente à estrutura *ArchiveTimeStampChain* existente. Dentro de uma mesma estrutura *ArchiveTimeStampChain*, todas as estruturas *ArchiveTimeStamp* devem empregar o mesmo algoritmo de resumo criptográfico, indicado no campo *digestAlgorithm* de cada estrutura *ArchiveTimeStamp*. Adicionalmente, pode-se criar uma nova árvore de resumos criptográficos a partir dos resumos criptográficos do campo *timeStamp* de várias estruturas *ArchiveTimeStamp* (BRANDNER; PORDESCH; GONDROM, 2007).

5.4.3 Renovação da Árvore de Resumo Criptográfico

Os algoritmos de resumo criptográfico tornam-se fracos ao longo dos anos, assim como qualquer mecanismo computacionalmente seguro, permitindo encontrar colisões e, posteriormente, pré-imagens. O registro de evidência permite a troca dos algoritmos por meio da renovação da árvore de resumo criptográfico (*Hash Tree Renewal*). Tal renovação deve ocorrer antes da quebra da resistência à segunda pré-imagem do algoritmo empregado.

A renovação da árvore de resumo criptográfico, com alguns ajustes para adaptá-la à proposta, tem os seguintes passos:

1. escolhe-se um algoritmo de resumo criptográfico H seguro;
2. calcula-se os resumo criptográficos $h(i)$ das testemunhas $d(i)$ referenciadas na estrutura *ArchiveTimeStamp* inicial obtendo-se $h(i) = H(d(i))$;
3. calcula-se os resumos criptográficos $ha(i)$ da concatenação de todas as estruturas *ArchiveTimeStampChain* $atsc(i)$ relacionadas a testemunha $d(i)$ obtendo-se $ha(i) = H(atsc(i))$;
4. calcula-se os resumos criptográficos $h(i)'$ da concatenação de cada $h(i)$ com $ha(i)$ obtendo-se $h(i)' = H(h(i)|ha(i))$;
5. gera-se uma nova estrutura *ArchiveTimeStamp* para cada $h(i)'$ ($h(i)'$ será tratado como o resumo das testemunhas (Ver seção 5.4.1);
6. gera-se uma nova estrutura *ArchiveTimeStampChain* contendo a estrutura *ArchiveTimeStamp* gerada e adiciona-se essa *ArchiveTimeStampChain* à estrutura *ArchiveTimeStampSequence* do registro de evidência.

Esta renovação era a principal dificuldade nas propostas visto que no segundo passo é necessário se obter o resumo criptográfico do objeto referenciado. A etapa de redistribuição modifica as partes do segredo e, assim, ter-se-ia um resumo criptográfico diferente que falharia na verificação do registro de evidência. A utilização das testemunhas tornou possível esta renovação uma vez que, empregando os mesmos gerador e módulos, tem-se sempre o mesmo resultado no corpo finito pois o segredo (o documento) é o mesmo.

5.4.4 Verificação do Registro de Evidência

A verificação do registro de evidência é uma prova não-repudiável que o documento existiu em um dado instante de tempo. Este procedimento deve ser executado pelo cliente sempre que recuperar o documento para se certificar da autenticidade do mesmo.

Do mesmo modo, adaptando a descrição do ERS, tem-se os seguintes passos:

1. verifica-se a estrutura *ArchiveTimeStamp* da primeira estrutura *ArchiveTimeStampChain* pelo resumo criptográfico da testemunha;

2. verifica-se cada estrutura *ArchiveTimeStampChain*: o primeiro resumo de cada estrutura *ArchiveTimeStamp* deve conter o resumo criptográfico do carimbo do tempo da estrutura *ArchiveTimeStamp* anterior. Cada carimbo do tempo deve ser válido para o tempo relativo ao carimbo sucessor. Todas as estruturas *ArchiveTimeStamp* da mesma estrutura *ArchiveTimeStampChain* devem usar o mesmo algoritmo de resumo criptográfico e este deve ser seguro no instante de tempo do carimbo do tempo da estrutura *ArchiveTimeStampChain* seguinte;

3. verifica-se a primeira lista de resumos criptográficos da primeira estrutura *ArchiveTimeStamp* de todas as outras estruturas *ArchiveTimeStampChain* pelo resumo criptográfico da concatenação do resumo da testemunha e do resumo de todas as estruturas *ArchiveTimeStampChain* anteriores. Verifica-se pelo último carimbo do tempo se foi gerado antes do último carimbo da estrutura *ArchiveTimeStampChain* tornar-se inválido.

A última verificação que deve ser realizada é em relação ao último carimbo do tempo que deve ser válido na data corrente, ou seja, deve ser verificada a assinatura digital desse carimbo.

Igualmente, o diferencial é a utilização das testemunhas como representação do documento sigiloso. Tal aplicação permite verificar a autenticidade de um documento sigiloso sem a necessidade de reconstruí-lo.

5.5 CONCLUSÃO

Elaborou-se dois protocolos para a preservação do sigilo e autenticidade de documentos eletrônicos a partir dos conceitos e definições, dos trabalhos relacionados e das análises das técnicas criptográficas a longo prazo e ataques as técnicas, abordados nos capítulos anteriores. Assim, preenche-se uma lacuna que havia na literatura, já que não existia uma proposta que tratasse do sigilo e autenticidade de documentos eletrônicos por longo prazo.

Propôs-se um protocolo inicial empregando parcialmente o protocolo G^2_{its}VSR e a sintaxe do registro de evidência, obtendo-se assim sigilo

e autenticidade de documentos eletrônicos. Todavia, tal proposta possui custos proibitivos e limitações que não aprimoram o que já existia na literatura, conforme descrito neste capítulo.

Desse modo, elaborou-se um novo protocolo, dessa vez utilizando integralmente o protocolo G^2_{its}VSR e incorporando integral e parcialmente diversos outros trabalhos, como Adams et al. (2001), Huhnlein et al. (2009), Desmedt e Jajodia (1997) e Pedersen (1991). Assim, acrescentou-se nos três procedimentos do protocolo G^2_{its}VSR novas funcionalidades, por exemplo a divisão do documento eletrônico em blocos, para que o protocolo provesse sigilo e autenticidade de documentos eletrônicos por longo prazo, mitigando os ataques baseados nos modelos de adversários estudados.

Também se adaptou os procedimentos do ERS para empregar o resumo das testemunhas, provenientes da técnica de compartilhamento de segredo verificável, ao invés do resumo do documento eletrônico em si. Essa alteração permitiu manter a autenticidade do documento mesmo que as partes sejam renovadas (modificadas), dada a relação matemática existente entre as partes do período anterior e atual.

No capítulo 6 o novo protocolo é avaliado de forma teórica e prática em relação ao que foi proposto. Além disso, faz-se uma discussão geral sobre a preservação de longo prazo.

6 AVALIAÇÃO E DISCUSSÃO DA PROPOSTA

6.1 INTRODUÇÃO

Este capítulo tem por objetivo avaliar e discutir o novo protocolo apresentado na seção 5.3.

Na seção 6.2 faz-se uma avaliação teórica desse protocolo em relação aos problemas e lacunas da arquitetura de referência apontados na seção 4.4.

A seguir, na seção 6.3, descreve-se as simulações e resultados obtidos a partir dos testes realizados com uma implementação parcial do protocolo descrito anteriormente.

Traz-se, também, uma discussão geral em relação a este trabalho, o protocolo proposto e a preservação de longo prazo de documentos eletrônicos na seção 6.4.

Por fim, faz-se as conclusões deste capítulo na seção 6.5.

6.2 AVALIAÇÃO TEÓRICA

A Arquitetura de Referência proposta por Huhnlein et al. (2009), descrita na seção 3.8, propôs uma infraestrutura para a preservação da autenticidade e do sigilo de documentos eletrônicos por longo prazo por meio da sintaxe do registro de evidência (ERS) e o compartilhamento de segredo, respectivamente. A partir da avaliação das técnicas de confidencialidade analisou-se a proposta da arquitetura de referência, com base a literatura científica. Como mostrado na seção 4.4 diversos problemas e lacunas não foram tratados nessa proposta.

Com base nos trabalhos de Huhnlein et al. (2009), Herzberg, Krawczyk e Yung (1995), Desmedt e Jajodia (1997) e Gupta e Gopinath (2007), entre outros, elaborou-se uma proposta para a preservação da autenticidade e do sigilo de documentos eletrônicos por longo prazo, descrita na seção 5.3. Nesta seção avalia-se essa proposta em relação aos problemas apontados na arquitetura de referência (ver seção 4.4).

A adoção do protocolo G^2_{its}VSR trata dos problemas de renovação das partes (*Share Renewal*), tolerância a LTAs maliciosos, adversários ativos e móveis, além de definir o protocolo para interação entre os participantes (servidores e clientes). A redistribuição do segredo renova as partes do segredo sem ser necessário reconstruí-lo, suporta mudanças na estrutura de acesso e compõe o mecanismo de defesa contra adversários móveis. A tolerância a LTAs maliciosos é provida por mecanismos de interação e execução do protocolo, baseado na maioria honesta, tolerando até $m-1$ LTAs maliciosos. A defesa contra adversários ativos vem dos canais de comunicação privados e da técnica de VSS adotada (PE-

DERSEN, 1991). Com a resolução de tais problemas obtém-se o sigilo da informação.

Ressalta-se, porém, que no protocolo G^2_{its}VSR e nos demais protocolos da família VSR, não é descrito um passo fundamental no procedimento de redistribuição, que foi adicionado ao protocolo proposto, de remoção dos pares do procedimento anterior (inicial ou redistribuição) e temporário (gerados durante a redistribuição). Sem essa etapa um adversário móvel poderia obter essas partes antigas ou temporárias, faltantes para m, e reconstruir o segredo. Tal remoção consta no trabalho de Desmedt e Jajodia (1997) e também de Herzberg, Krawczyk e Yung (1995).

Outro problema impactante da Arquitetura de Referência é quanto à renovação da árvore de resumo criptográfico (*Hash Tree Renewal*). Como visto, qualquer mudança na estrutura de acesso ou se usado o modo de compartilhamento interno (*Inside Shared Mode*) levaria à reconstrução do segredo e consequente alteração na árvore de Merkle. Também foi proposta construção de uma nova árvore nesses casos, o que não é suportado pelo ERS. Em ambos os casos perder-se-ia o histórico dos objetos necessários para o ERS. Ainda, foi proposta a obtenção de resumos criptográficos com diferentes algoritmos para evitar a reconstrução, mas isso apenas posterga o problema. Tais problemas foram solucionados empregando as testemunhas das partes do segredo ao invés das partes em si, visto que, como o segredo não se altera, o produtório das testemunhas resulta sempre no mesmo valor. Tal característica é necessária para utilização do ERS. Essa solução provê autenticidade à informação.

O emprego do ERS ainda provê escalabilidade visto que, para um mesmo carimbo do tempo, pode-se agregar tantas testemunhas quanto desejado. Para cada testemunha agregada em uma mesma árvore de Merkle ganha-se, aproximadamente, o espaço em disco que ocuparia um carimbo do tempo. Aproximadamente pois existe o uso de disco para armazenar o resumo criptográfico dessa testemunha e a estrutura do ERS. Supondo que um carimbo do tempo ocupe c bytes de armazenamento em disco, para 100 documentos ter-se-ia o uso de $100.c$ bytes em disco, se fossem empregados carimbos individuais para esses documentos. Já com o uso do ERS, tem-se o uso de $100 + c$ bytes em disco, despesando-se, em ambos os casos, os custos de armazenamento da estrutura de dados empregada.

Tem-se ainda duas lacunas remanescentes: o monitoramento e manutenção das partes armazenadas e o gerenciamento do identificador (ID) do documento por parte do cliente.

6.3 SIMULAÇÃO E RESULTADOS

Implementou-se como prova de conceito o fluxo principal do protocolo xVSR (GUPTA; GOPINATH, 2006), de forma centralizada. Adaptou-se o protocolo, conforme descrito por Huhnlein et al. (2009), para manipulação de documentos eletrônicos.

Verificou-se a execução do protocolo empregando a técnica de

Feldman (1987), por meio de um exemplo, com o compartilhamento de um pequeno documento eletrônico (12 KiB). Constatou-se também que, mesmo após diversas redistribuições, obteve-se sempre os mesmos valores de testemunhas, confirmando a aplicabilidade das testemunhas no ERS.

Por meio do exemplo da seção 3.2 percebeu-se uma associação entre a quantidade de redistribuições possíveis e o corpo finito utilizado. Naquele exemplo, o módulo empregado era $p = 53$ e $m = 2$, resultando em uma função de primeiro grau. Como o valor dos coeficientes desse polinômio estão compreendidos entre 1 e 52 (desprezando-se o 0 pois, assim, ter-se-ia a exposição do segredo), tem-se um máximo de 52 compartilhamentos possíveis para um determinado segredo. Para $m \geq 2$, considerando-se que pelo menos um dos coeficientes deve ser diferente de zero (de modo a não expor o segredo), tem-se $p^{m-1} - 1$ combinações de coeficientes possíveis. Logo, essa é a quantidade máxima de redistribuições para um determinado coeficiente.

Após esgotar o número de execuções os coeficientes começarão a se repetir e, assim, ter-se-á as mesmas partes do segredo. A primeira vista o esgotamento dos coeficientes parece tornar inviável o emprego do protocolo. Entretanto, a medida que o valor do módulo e/ou o número mínimo de servidores aumenta, aumenta também as combinações de coeficientes possíveis. Por exemplo, o número primo[1] $2^{48} - 257$, de apenas 48 bits, empregado em um compartilhamento com $m = 2$ e $n = 3$, mesmo se houvesse uma redistribuição por hora, seria suficiente para mais de 32 bilhões de anos.

Outra constatação diz respeito à necessidade de controle dos coeficientes utilizados no compartilhamento, uma vez que os mesmos valores de coeficientes geram as mesmas partes para o mesmo segredo. Se um adversário estiver monitorando e obtendo tantas partes quanto possível, em um caso de repetição, apesar de uma baixa probabilidade de ocorrência, esse adversário poderá obter as partes faltantes para m e reconstruir o documento sigiloso.

Outro teste realizado foi quanto a mudanças na estrutura de acesso no procedimento de redistribuição. Nesse teste fez-se uma distribuição inicial com $m = 3$ e $n = 5$. Executou-se uma redistribuição, considerando que dois dos servidores não participaram – simulando uma indisponibilidade, tendo-se o mínimo de servidores necessários para execução do protocolo, resultando em uma estrutura de acesso ($m = 3, n = 3$). Nesse cenário, três dos servidores tiveram suas partes do segredo atualizadas enquanto que os outros dois mantiveram suas partes antigas e, no período atual, inválidas. Na redistribuição seguinte considerou-se o total de servidores novamente, distribuindo as partes da estrutura de acesso ($m = 3, n = 3$) novamente para ($m = 5, n = 3$). Verificou-se que todos os servidores terminaram a execução com uma nova parte do segredo, todas diferentes entre si. Ou seja, desde que se mantenha o número mínimo

[1] Obtido em: <http://primes.utm.edu/lists/2small>. Verificado com o teste de Miller-Rabin.

de servidores necessários (m), pode-se mudar a estrutura de acesso conforme desejado ou necessário. Com esse teste verificou-se que a falta de até $m-1$ servidores é suportada pelo protocolo, conforme descrito pelos autores.

Em relação ao procedimento inicial e o espaço em disco utilizado, a fórmula 5.2 vista na seção 5.2.1 é a mesma. Tem-se que $c = b \cdot n \cdot t$, sendo b o número de blocos do documento, t o tamanho (em bytes) de cada bloco e n o total de servidores.

Entretanto, a fórmula para o cálculo do espaço em disco após um procedimento de redistribuição é diferente, visto a modificação realizada nas etapas do protocolo, que culminava em um crescimento de partes do segredo proporcional ao novo número de servidores. De fato, a quantidade de partes do segredo permanece constante no sistema e é igual ao novo número de servidores.

Assim, o espaço em disco em uso durante a redistribuição será dado por $c' = b \cdot n' \cdot t + b \cdot n \cdot t$. Como $b \cdot n \cdot t = c$, tem-se que $c' = c + b \cdot n' \cdot t$. O termo c somado à c' deve-se ao fato de que, durante a execução do protocolo, ter-se-á as partes do período anterior e as partes do período atual.

Todavia, após a verificação das novas partes por meio das testemunhas, tem-se a liberação de c bytes. Logo, tem-se que o custo de armazenamento após uma redistribuição é $c' = b \cdot n' \cdot t$.

Usou-se a variável n' pois o número de servidores pode ser alterado livremente, desde que $n' \geq m$. Os parâmetros b e t permanecem os mesmos visto que a alteração do módulo empregado no compartilhamento requer a alteração no módulo empregado no VSS o que, consequentemente, modificaria as testemunhas e, além de falhar a verificação no protocolo, resultaria em falha na verificação e na ligação do ERS com o documento sigiloso.

O compartilhamento de segredo exige que, para se obter segurança incondicional, as partes sejam pelo menos do mesmo tamanho do segredo. Assim, tem-se o uso do disco de pelo menos n vezes o tamanho d do documento eletrônico sigiloso. Logo, a adição de um servidor à infraestrutura aumenta o uso do disco em pelo menos mais d bytes. Nos testes realizados, usou-se o tamanho do módulo como parâmetro para o tamanho do bloco, logo o tamanho das partes não ultrapassa o tamanho do módulo (aritmética modular) e tem-se aproximadamente $n.d$ bytes de uso do disco. Com isso, a adição de cada novo servidor à infraestrutura aumenta o uso do disco em d bytes. Do mesmo modo, a remoção de cada servidor libera d bytes de espaço em disco.

6.4 DISCUSSÃO SOBRE A PROPOSTA

Nesta seção discute-se sobre a proposta, limitações remanescentes, manutenções e a preservação de longo prazo, apresentando-se sugestões de estratégias a serem adotadas.

Descreveu-se na seção 5.4 o emprego do resumo criptográfico das testemunhas de modo a representar as partes do documento preservado, visto que a relação matemática entre as partes permite obter as mesmas testemunhas para diferentes partes do mesmo segredo. Todavia, tem-se uma redução na otimização conseguida pelo ERS. Enquanto um ERS continha o resumo de vários documentos na proposta este contém apenas um documento. De modo a otimizar esse resultado apresenta-se a seguir algumas ideias.

No novo protocolo descreveu-se que cada folha da árvore de Merkle seria o resumo criptográfico de uma testemunha, sendo que cada testemunha é referente a um bloco do documento. Para exemplificar, suponha que se tenha 100 testemunhas de um documento compartilhado. Assim, ter-se-ia 100 resumos criptográficos nas folhas, 50 no primeiro nível, e assim sucessivamente até chegar a raiz, resultando em 199 (ou $2 \cdot f - 1$, sendo f o número de folhas) resumos criptográficos a serem armazenados. Considerando o algoritmo SHA-1[2], que possui 160 bits, seriam necessários 31.840 bits ou 3.980 bytes de espaço em disco para armazenamento. Entretanto, ter-se-ia o cálculo de 199 resumos criptográficos.

Como todos os blocos são necessários para reconstrução do documento tem-se algumas otimizações possíveis. A utilização dos grupos de dados do ERS pode economizar cálculos de resumo criptográfico e alguns bytes de espaço em disco, agrupando um determinado número de testemunhas e obtendo-se o resumo desse grupo. Supondo que se agrupasse as testemunhas em grupos de 10, ter-se-ia 10 cálculos e resumos criptográficos, sendo necessário armazenar somente 1600 bits ou 200 bytes, quase 20 vezes menos, além de executar 10 vezes menos cálculos de resumo criptográfico, comparados à configuração padrão. Poder-se-ia criar um único grupo contendo todas as testemunhas e, assim, somente um resumo criptográfico seria calculado e armazenado. No exemplo anterior resultaria em 160 bits de armazenamento e somente um cálculo de resumo criptográfico.

Uma outra possibilidade seria efetuar operações de produto aritmético módulo r das testemunhas, com tantas testemunhas quanto desejado, obtendo-se os mesmos ganhos em relação à criação de grupos de dados do ERS.

O ERS gerado na execução do protocolo, mesmo com as otimizações descritas acima, ainda seria individual para cada documento, ou seja, uma assinatura digital, perdendo-se assim a vantagem de utilização do ERS. Contudo, comparado à quantidade de dados e complexidade de padrões como CAdES, XAdES e PAdES, ter-se-ia vantagem no uso do ERS ao invés dos referidos formatos. Ainda, conforme descrito no ERS, pode-se gerar novos registros de evidência a partir de outros registros. Assim, consegue-se obter a mesma granularidade resultante da aplicação do ERS em documentos.

O gerenciamento da infraestrutura compreende a manutenção das

[2] *Secure Hash Algorithm.*

propriedades que um serviço de arquivamento por longo prazo deve preservar indefinidamente. Tais propriedades, como autenticidade, confidencialidade e disponibilidade precisam de constante tratamento visto que o tempo é o maior inimigo de sistemas de arquivamento.

Faz-se necessária a manutenção da própria árvore Merkle (1980) para que a autenticidade (e integridade) seja mantida. Como visto na seção 3.6 a manutenção do registro de evidência compreende dois procedimentos: a renovação do carimbo do tempo e a renovação da árvore de resumo criptográfico.

Os carimbos do tempo nas raízes das árvores precisam ser renovados ativamente por meio da renovação do carimbo do tempo. Esse procedimento não requer qualquer acesso as informações arquivadas. Outra manutenção necessária, a renovação da árvore de resumo criptográfico, requer acesso a todas as testemunhas das partes relativas aos documentos arquivados para, então, reconstruir as árvores. Ambos os processos foram descritos na seção 5.4.

A autenticidade dos documentos eletrônicos arquivados está baseada nas árvores de Merkle, o que as tornam atrativas aos adversários. Por esse motivo deve-se manter replicações dessas árvores para evitar um ponto único de falha. Também, nesse sentido, diferentes algoritmos devem ser aplicados, evitando uma quebra abrupta que possa comprometer o sistema. Assim, sugere-se que cada servidor utilize conjuntos de algoritmos e tenha certificados de autoridades distintas. Desse modo, mesmo que um algoritmo venha a ser quebrado ou uma autoridade seja comprometida ou revogada, ter-se-á outras árvores de redundância.

O ERS garante a autenticidade das partes que compõem os documentos eletrônicos preservados. Todavia, os dados escritos nas mídias digitais podem ser alterados ou removidos, sendo esse um dos problemas remanescentes da proposta. Assim, um monitoramento e manutenção da integridade das partes armazenadas é necessária, conforme descrito na seção 4.4.

Tal monitoramento da integridade nos servidores pode se dar por meio de resumos criptográficos armazenados (STORER et al., 2009), com o intuito de diminuir a probabilidade de alterações errôneas ou maliciosas nas partesdo segredo. Além disso, pode-se utilizar técnicas de espelhamento, redundância e/ou códigos de correção de erros (*erasure codes*), tanto locais quanto distribuídas. Informações e referências para tais técnicas podem ser obtidas em Patterson et al. (1989), Stonebraker e Schloss (1990) e Chang et al. (2002).

Deve-se monitorar os algoritmos utilizados na infraestrutura em relação à segurança provida e, antes de tornarem-se inseguros, deve-se substituí-los por outros considerados seguros, invocando os procedimentos necessários.

A disponibilidade dos documentos eletrônicos arquivados depende da disponibilidade de pelo menos m servidores na infraestrutura. Contudo, problemas locais e de conexão podem interromper a disponibilidade

dessas informações. Sugere-se, assim, a redundância do sistema em relação ao *link* de Internet e também quanto à distribuição geográfica, sendo desejável ter pelo menos uma replicação em localidade distinta da principal, atuando como um espelho.

A segurança contra adversários móveis reside na redistribuição dos segredos antes que um adversário obtenha as m partes necessárias para reconstrução do segredo, ou seja, do documento eletrônico. Definir uma política de redistribuição não é um processo trivial visto que não há um modo preciso de determinar se houve um comprometimento e quais informações foram acessadas. Entretanto, viu-se nas simulações e testes realizados que mesmo módulos de 48 bits são suficientes para redistribuições constantes. As políticas de redistribuição devem levar em conta os valores de limiar, incidentes de intrusão, o número de partes que podem ser comprometidas ($m - 1$), a indisponibilidade de servidores, entre outros. Sugere-se revisar essa política de acordo com tais variáveis.

O último problema remanescente é o gerenciamento de IDs – identificadores dos objetos arquivados, por parte dos clientes. Essa abordagem é interessante de modo que não é preciso ter uma lista centralizada desses arquivos, o que constitui um ponto a ser atacado no sistema – como um servidor central de recuperação (ver seção 4.2.2).

Entretanto, clientes não possuem infraestrutura para guardar tal informação por longo prazo. O sistema operacional do cliente pode ser formatado ou ter um problema de hardware – comumente não há cópia de segurança. Se armazenado em *tokens* criptográficos o mesmo pode ser roubado, perdido ou mesmo vir a ter problemas. Em Storer et al. (2009), utilizou-se a técnica de *approximate pointes* a qual utiliza um encadeamento de ponteiros que dificulta um adversário encontrar a localização do documento. Contudo, caso o cliente perca o identificador, o sistema pode encontrá-lo.

Em uma outra abordagem pode-se aproveitar a distribuição do sistema mantendo com cada servidor uma cópia da lista que identifica os documentos do cliente. Assim, o cliente poderia obter a lista pela maioria dos servidores.

Igualmente, pode-se utilizar o compartilhamento de segredo de modo a distribuir a lista entre todos os servidores. Quando um cliente acessar o sistema, a lista é então reconstruída para, então, indexar os documentos do cliente.

O presente trabalho não possui foco nos aspectos gerais do arquivamento de longo prazo e, portanto, não direcionou a pesquisa para tais aspectos. Todavia, pelos conhecimentos obtidos em trabalhos anteriores, como Silva e Ramos (2007), pode-se sugerir algumas estratégias e aspectos gerais da preservação de longo prazo.

Em relação a obsolescência de mídias o emprego de técnicas de redundância – como RAID (PATTERSON et al., 1989), e o uso de diferentes tecnologias de mídias – como discos de estado sólidos e discos rígidos, além do próprio limiar do compartilhamento de segredo, adicio-

nam redundância suficientes para mitigar essa classe de problemas.

A obsolescência de formatos já é um problema mais complexo. Entretanto, sugere-se a adoção de padrões abertos de forma a não comprometer a legibilidade dos documentos arquivados.

No tocante as estratégias de preservação, sugere-se empregar a emulação e, em casos extremos, a notarização. A estratégia de emulação permite emular softwares em plataformas distintas (portabilidade), evitando assim a migração de formato. A notarização seria a última opção, utilizando-se de um notário para atestar a autenticidade das informações migradas. No caso de a emulação não ser suficiente, será necessária a participação do cliente para reconstrução do documento, visto que os servidores não devem ter autonomia para realizar essa operação.

6.5 CONCLUSÃO

Neste capítulo avaliou-se teoricamente o novo protocolo em relação aos problemas e lacunas da arquitetura de referência, visto essa ser a única referência em sigilo e autenticidade de documentos eletrônicos de forma conjunta. Nessa avaliação verificou-se que o protocolo proposto suprime os principais problemas dessa arquitetura, deixando duas lacunas em aberto: o monitoramento e manutenção das partes armazenadas e o gerenciamento do identificador (ID) do documento por parte do cliente. Nas discussões tais lacunas foram abordadas e apresentou-se algumas alternativas.

Descreveu-se, também, as simulações e resultados obtidos a partir dos testes realizados com uma implementação parcial do protocolo proposto. Apesar de limitada, com tal implementação pode-se verificar a execução do protocolo e obter-se diversos dados que foram apresentados. Em especial, mostrou-se a questão do esgotamento do espaço modular e a necessidade de controle na repetição dos coeficientes aleatórios, que não foram comentados na literatura, provavelmente pela baixa probabilidade de ocorrência dado um módulo de tamanho razoável, conforme descrito.

Por fim, discutiu-se de modo geral em relação a este trabalho, o protocolo proposto e à preservação de longo prazo de documentos eletrônicos, complementando com alguns itens não abordados.

7 CONSIDERAÇÕES FINAIS

Este trabalho apresentou dois protocolos para a preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo. O protocolo preliminar possui limitações e custo elevado, que além de torná-lo impraticável, não representa um avanço significativo em relação a literatura científica. Já o novo protocolo, elaborado a partir do preliminar, aprimorou os resultados obtidos e corrigiu as limitações presentes na proposta anterior. Tal protocolo contempla o requerido no objetivo geral.

No capítulo 2 definiu-se os conceitos de segurança utilizados ao longo deste trabalho. Listou-se os princípios de Kerckhoffs, os quais norteiam o desenvolvimento de técnicas criptográficas modernas. Realizou-se um levantamento das técnicas aplicadas à preservação do sigilo de documentos eletrônicos existentes na literatura científica e tecnológica, além dos modelos de ataque considerados para essas técnicas. Igualmente, descreveu-se as técnicas de assinatura digital e carimbo do tempo empregados como forma de preservação da autenticidade por longo prazo.

Com base no conhecimento dessas técnicas, buscou-se os trabalhos relacionados disponíveis na literatura que empregassem tais técnicas, aprofundando-se nas abordagens mais relevantes a este trabalho. Verificou-se a existência de uma proposta unindo sigilo e autenticidade – a arquitetura de referência de Huhnlein et al. (2009). Essas informações estão contempladas no capítulo 3.

Com o levantamento das técnicas e verificação dos trabalhos relacionados, no capítulo 4 avaliou-se a aplicabilidade destas técnicas em longo prazo, de acordo com a literatura científica, comparando seus benefícios e limitações. Discutiu-se, também, os modelos de adversários aos quais as técnicas eram suscetíveis, uma vez que determinado ataque pode comprometer a preservação de longo prazo, o que de fato comprovou-se. Por fim, avaliou-se a arquitetura de referência – trabalho pioneiro na união de sigilo e autenticidade com as características descritas – em relação às técnicas, aplicabilidade e modelos de adversários discutidos neste capítulo.

A partir dos estudos realizados anteriormente e da verificação de diversos problemas e lacunas na arquitetura de referência, com base nos trabalhos relacionados e a avaliação das técnicas e ataques por longo prazo, propôs-se no capítulo 5 dois protocolos para a preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo. Tais protocolos incorporam diversos mecanismos e técnicas vistos na literatura científica de modo a adequá-los à preservação de longo prazo.

Ainda, descreveu-se duas equações para o cálculo aproximado dos custos de armazenamento após os procedimentos inicial e de redistribuição, de modo a permitir uma melhor avaliação do protocolo preliminar. Verificou-se que, apesar de possível, essa proposta era impraticável e pos-

suía limitações quanto aos objetivos propostos, avançando em relação a arquitetura de referência, mas não muito diferente em relação a propostas constantes na literatura, por exemplo, empregando a renovação criptográfica. Assim, baseado no protocolo preliminar, propôs-se um novo protocolo e a integração deste com a sintaxe do registro de evidência por meio do resumo criptográfico das testemunhas, obtidas na execução da técnica de compartilhamento de segredo verificável, ao invés do resumo do documento em si. Essa proposta permitiu a renovação das partes do segredo sem reconstrução e de forma distribuída.

Avaliou-se no capítulo 6 o novo protocolo de modo a verificá-lo em relação aos problemas existentes. Tal avaliação abrangeu as técnicas empregadas por longo prazo, os modelos de atacantes considerados, a arquitetura de referência e suas deficiências. Além disso, verificou-se, por meio de um implementação simplificada, a execução das técnicas e mecanismos propostos incorporados que formaram o protocolo proposto. Com base nessa implementação simulou-se e testou-se diversas situações e configurações. Os testes executados permitiram a conclusão de alguns pontos que não foram discutidos na literatura científica, nos trabalhos pesquisados, como a questão do espaço modular e o controle dos coeficientes.

Também neste capítulo elaborou-se duas fórmulas para o cálculo aproximado dos custos de armazenamento, após os procedimentos inicial e de redistribuição para o novo protocolo. Comparativamente ao protocolo preliminar, houve uma melhora proporcional a $n \cdot t$ vezes nos custos de armazenamento, sendo n o número de servidores e t o número de redistribuições executadas, considerando que a mesma estrutura de acesso seja mantida. No protocolo preliminar a cada período de redistribuição geram-se $n.n'$ partes do segredo, sendo n' o novo número de servidores, enquanto que no novo protocolo tem-se n' partes. Projetando-se que o número de servidores tende a aumentar ao invés de diminuir, percebe-se que o protocolo preliminar tem custos impraticáveis.

Por fim, discutiu-se sobre o protocolo proposto, manutenções necessárias às técnicas e mecanismos empregados e outros aspectos relevantes à preservação de longo prazo. Além disso, descreveu-se algumas ideias para problemas remanescentes, estratégias de preservação e variações de configuração do protocolo preliminar e da sintaxe do registro de evidência de modo a otimizar tais propostas.

Por meio dos parágrafos acima verifica-se a abrangência e cumprimento dos objetivos geral e específicos propostos no capítulo introdutório deste trabalho.

Em resumo, as contribuições deste trabalho são: pesquisa das técnicas que promovem o sigilo da informação; condensação dos modelos de adversários aos protocolos; aprofundamento em trabalhos relacionados à confidencialidade e autenticidade; avaliação das técnicas pesquisadas em relação ao nível de segurança que promovem e aos modelos de adversário; avaliação da arquitetura de referência com base nas avaliações das técnicas e modelos de adversários; proposição de dois protocolos para pre-

servação do sigilo e autenticidade de documentos eletrônicos por longo prazo; avaliação dos protocolos propostos em relação as deficiências da arquitetura de referência; e discussão sobre os protocolos propostos, recomendações adicionais, problemas remanescentes e a preservação de longo prazo.

Todavia, tem-se algumas limitações quanto ao protocolo proposto. Além dos trabalhos futuros descritos no parágrafo seguinte, que são importantes para complementar a proposta, cita-se o gerenciamento da identificação dos documentos preservados e a preservação da integridade dos dados armazenados. Essas limitações foram discutidas na seção 6.4 e algumas ideias foram descritas.

A diversidade de áreas e especialidades envolvidas na elaboração desta dissertação deixa espaço para trabalhos futuros. Sugere-se, entre outras possibilidades: a modelagem e verificação formal do protocolo para atestar sua confiabilidade; a busca por heurísticas para escolha do limiar e do período ótimo para redistribuição; a proposta de uma arquitetura que empregue o protocolo proposto; o cálculo do tamanho das partes do compartilhamento por meio da teoria da informação; a incorporação de mecanismos para autenticação por longo prazo; um estudo e definições das estratégias de preservação a fim de mitigar a obsolescência das tecnologias envolvidas; implementar o protocolo em sua plenitude; e procurar e verificar os componentes, técnicas e algoritmos empregados para aprimorar o desempenho do protocolo.

Desse modo, conclui-se este trabalho tendo como principal contribuição um protocolo para preservação do sigilo e autenticidade de documentos eletrônicos por longo prazo baseado em compartilhamento de segredo, renovação criptográfica sem reconstrução do segredo, assinaturas digitais, carimbos do tempo e sintaxe do registro de evidência.

REFERÊNCIAS

ADAMS, C. et al. Internet X.509 Public Key Infrastructure Time-Stamp Protocolo (TSP). *Internet Engineering Task Force (IETF) Networking Group, Request for Comments*, RFC 3161, 2001.

ANDERSON, R.; PETITCOLAS, F. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, v. 16, n. 4, p. 474–481, maio 1998. ISSN 07338716.

BLAKLEY, G. Safeguarding cryptographic keys. *International Workshop on Managing Requirements Knowledge*, v. 0, p. 313, 1979.

BRANDNER, R.; PORDESCH, U. Long-term conservation of provability of electronically signed documents. *Beitrag zu ISSE*, p. 2–5, 2002.

BRANDNER, R.; PORDESCH, U.; GONDROM, T. Evidence Record Syntax (ERS). *Internet Engineering Task Force (IETF) Networking Group, Request for Comments*, v. 4998, 2007.

BRASIL. *Lei nº 5.172, de 25 de outubro de 1966*. Poder Executivo, Brasília, DF, 1966.

BRASIL. *Lei nº 5.869, de 11 de janeiro de 1973*. Poder Executivo, Brasília, DF, 1973.

BRASIL. *Medida provisória nº 2.200-2, de 24 de agosto de 2001*. Poder Executivo, Brasília, DF, 2001.

BRASIL. *Lei nº 11.419, de 19 de dezembro de 2006*. Poder Executivo, Brasília, DF, 2006.

CHANG, F. et al. *Myriad: Cost-effective Disaster Tolerance*. [S.l.]: In Proceeedings of FAST, 2002. 103–116 p.

CHOR, B. et al. *Verifiable secret sharing and achieving simultaneity in the presence of faults*. [S.l.]: IEEE, 1985. 383–395 p. ISBN 0-8186-0644-4.

CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS. *Reference Model for an Open Archival Information System (OAIS)*. 2002.

CUSTóDIO, R. F. et al. Temporal Key Release Infrastructure. In: *6th Annual PKI R&D Workshop*. Gaithersburg, MD: [s.n.], 2007.

DENNING, D. *Cryptography and data security*. [S.l.]: Addison-Wesley, 1982. ISBN 0-201-10150-5.

DESMEDT, Y.; JAJODIA, S. *Redistributing secret shares to new access structures and its applications*. 1997. 1–14 p.

Di Lucca, G. et al. Identifying cross site scripting vulnerabilities in web applications. In: *Web Site Evolution, 2004. WSE 2004. Proceedings. Sixth IEEE International Workshop on*. [S.l.]: IEEE, 2005. p. 71–80. ISBN 0769522246. ISSN 1550-4441.

DIAS, J. S. *Confiança no Documento Eletrônico*. 141 f. Tese (Doutorado em Engenharia de Produção) — Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2004.

DIFFIE, W.; HELLMAN, M. New directions in cryptography. *IEEE Transactions on information Theory*, v. 22, n. 6, p. 644–654, 1976. ISSN 0018-9448.

EUROPEAN NETWORK OF EXCELLENCE IN CRYPTOLOGY II. *ECRYPT II Yearly Report on Algorithms and Keysizes (2009-2010)*. [S.l.], Mar 2010.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)*. [S.l.], Nov 2009.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term – PAdES-LTV Profile*. [S.l.], Jul 2009.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)*. [S.l.], Jun 2009.

FELDMAN, P. A practical scheme for non-interactive verifiable secret sharing. *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, Ieee, p. 427–438, out. 1987.

GUPTA, V.; GOPINATH, K. *An Extended Verifiable Secret Redistribution Protocol for Archival Systems*. [S.l.]: IEEE, 2006. 100–107 p. ISBN 0-7695-2567-9.

GUPTA, V. H.; GOPINATH, K. G^2_{its}VSR: An Information Theoretical Secure Verifiable Secret Redistribution Protocol for Long-term Archival Storage. *Fourth International IEEE Security in Storage Workshop*, Ieee, p. 22–33, 2007.

HABER, S.; STORNETTA, W. How to time-stamp a digital document. *Journal of Cryptology*, v. 3, n. 2, p. 99–111, jan. 1991. ISSN 0933-2790.

HE, J.; DAWSON, E. Shared secret reconstruction. *Designs, Codes and Cryptography*, Springer, v. 14, n. 3, p. 221–237, 1998.

HERZBERG, A.; KRAWCZYK, H.; YUNG, M. Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. *IBM TJ Watson Research Center*, p. 1–22, 1995.

HUHNLEIN, D. et al. A Comprehensive Reference Architecture for Trustworthy Long-Term Archiving of Sensitive Data. *2009 3rd International Conference on New Technologies, Mobility and Security*, Ieee, p. 1–5, dez. 2009.

INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. *Visão Geral do Sistema de Carimbos do Tempo na ICP-Brasil*. v. 1.2. Brasília, Abril 2010. DOC-ICP-11.

INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. *Visão Geral Sobre Assinaturas Digitais na ICP-Brasil*. v. 2.0. Brasília, Abril 2010. DOC-ICP-15.

INTERNATIONAL STANDARDIZATION ORGANIZATION. *ISO/IEC 14721:2003: Space Data and Information Transfer Systems — Open Archival Information System — Reference Model*. Geneva, Switzerland, 2003. 141 p.

JAMIL, T. Steganography: the art of hiding information in plain sight. *Potentials, IEEE*, IEEE, v. 18, n. 1, p. 10–12, 1999. ISSN 0278-6648.

KERCKHOFFS, A. La cryptographie militaire. *Journal des sciences militaires*, vol. IX, 1883.

MAURER, U. Information-theoretic cryptography. In: *Advances in Cryptology—CRYPTO'99*. [S.l.]: Springer, 1999. p. 785–785.

MAURER, U. M. *The Role of Information Theory in Cryptography*. [S.l.]: In Fourth IMA Conference on Cryptography and Coding, 1993. 49–71 p.

MAY, T. C. *Timed-Release Crypto*. [S.l.], 1993.

MENEZES, A.; OORSCHOT, P. V.; VANSTONE, S. *Handbook of applied cryptography*. [S.l.]: CRC, 1997.

MERKLE, R. C. Protocols for public key cryptosystems. *Security and Privacy, IEEE Symposium on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 122, 1980. ISSN 1540-7993.

NIKOV, V.; NIKOVA, S. *On Proactive Secret Sharing Schemes*. 2005. 308–325 p.

NOTOYA, A. E. *IARSDE*: Infra-estrutura de armazenamento e recuperação segura de documentos eletrônicos. 110 f. Dissertação (Mestrado em Ciência da Computação) — Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2002.

PASQUAL, E. S. *IDDE*. 110 f. Dissertação (Mestrado em Ciência da Computação) — Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2001.

PATTERSON, D. et al. Introduction to redundant arrays of inexpensive disks (RAID). In: *COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers*. [S.l.]: IEEE Comput. Soc. Press, 1989. p. 112–117. ISBN 0-8186-1909-0.

PEDERSEN, T. Non-interactive and information-theoretic secure verifiable secret sharing. In: *Crypto*. [S.l.]: Springer, 1991. v. 91, p. 129–140.

RABIN, M. O. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM (JACM)*, v. 36, n. 2, p. 335, 1989. ISSN 0004-5411.

RAMOS, T. A. et al. An infrastructure for long-term archiving of authenticated and sensitive electronic documents. In: CAMENISCH, J.; LAMBRINOUDAKIS, C. (Ed.). *EuroPKI 2010*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2011. (Lecture Notes in Computer Science, v. 6711), p. 193–207.

RIVEST, R.; SHAMIR, A.; WAGNER, D. *Time-lock puzzles and timed-release crypto*. Cambridge, MA, USA: Massachusetts Institute of Technology Cambridge, MA, USA, 1996.

RIVEST, R. L.; SHAMIR, a.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, v. 21, n. 2, p. 120–126, fev. 1978. ISSN 00010782.

SCHNEIER, B. *Applied Cryptography (Second Edition)*. [S.l.]: John Wiley & Sons, 1996.

SHAMIR, A. How to share a secret. *Communications of the ACM*, ACM, v. 22, n. 11, p. 612–613, 1979.

SHANNON, C. E. Communication Theory of Secrecy Systems. *M.D. computing : computers in medical practice*, v. 15, n. 1, p. 57–64, 1948. ISSN 0724-6811.

SHIREY, R. *Internet Security Glossary, Version 2*. [S.l.]: IETF, ago. 2007. RFC 4949 (Informational). (Request for Comments, 4949).

SILVA, N. da et al. *Central Notarial de Serviços Eletrônicos Compartilhados*. São Caetano do Sul, SP: Yendis Editora, 2007.

SILVA, N. da; RAMOS, T. A. *Preservação de Longo Prazo de Documentos Eletrônicos na CNSEC*. 130 f. Monografia (Bacharelado em Ciência da Computação) — Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2007.

SIMMONS, G. *Contemporary Cryptology: The Science of Information integrity*. 1. ed. [S.l.]: Wiley-IEEE Press, 1999. 656 p. ISBN 9780470544327.

STALLINGS, W. *Cryptography and network security: principles and practice*. Third. [S.l.]: Prentice-Hall, Inc., 2002. 696 (est.) p.

STONEBRAKER, M.; SCHLOSS, G. A. *Distributed Raid – A New Multiple Copy Algorithm*. [S.l.]: IEEE Press, 1990. 430–437 p.

STORER, M. W.; GREENAN, K.; MILLER, E. L. Long-term threats to secure archives. *Proceedings of the second ACM workshop on Storage security and survivability - StorageSS '06*, ACM Press, New York, New York, USA, p. 9, 2006.

STORER, M. W. et al. POTSHARDS—a secure, recoverable, long-term archival storage system. *ACM Transactions on Storage*, v. 5, n. 2, p. 1–35, 2009. ISSN 15533077.

WANG, E. et al. *A Key-Recovery System for Long-term Encrypted Documents*. [S.l.]: IEEE, 2006. 52–52 p. ISBN 0-7695-2743-4.

WONG, T.; WANG, C.; WING, J. Verifiable secret redistribution for archive systems. *First International IEEE Security in Storage Workshop, 2002. Proceedings.*, IEEE Comput. Soc, n. December, p. 94–105, 2002.

ZIMMER, W.; LANGKABEL, T.; HENTRICH, C. ArchiSafe: Legally Compliant Electronic Storage. *IT Professional*, v. 10, n. 4, p. 2633, 2008.